AFRL-IF-RS-TR-2005-415
Final Technical Report
January 2006

MULTI-AGENT FRAMEWORK FOR THE FAIR DIVISION OF RESOURCES AND TASKS

Metron Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K543

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-415 has been reviewed and is approved for publication

APPROVED: /s/

JOSEPH V. BEASOCK
Project Engineer

FOR THE DIRECTOR: /s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** JANUARY 2006	**3. REPORT TYPE AND DATES COVERED** Final Jun 00 – Jan 05

4. TITLE AND SUBTITLE
MULTI-AGENT FRAMEWORK FOR THE FAIR DIVISION OF RESOURCES AND TASKS

5. FUNDING NUMBERS
C - F30602-00-C-0175
PE - 62301E
PR - TASK
TA - 00
WU - 02

6. AUTHOR(S)
Gregory A. Godfrey

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Metron, Incorporated
11911 Freedom Drive
Suite 800
Reston Virginia 20190-5602

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFSA
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-415

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Joseph V. Beasock/IFSA/(315) 330-3051/ Joseph.Beasock@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

Under this contract, we developed technology that addresses the dynamic problem of autonomous, competitive agents negotiating over the fair division of resources and tasks. We have applied this multi-agent technology to two military domains: (1) commercial airlift procurement for large contingencies and (2) unmanned aerial vehicle (UAV) coordinated search and surveillance. The collaborative auction and mission exchange approach that we developed makes planning more flexible, missions more reliable, and leverages commercial operational “best practices” without having to integrate those practices into military systems or to make the expertise available to competitors. The UAV challenge is achieving real-time, effective coordination of a fleet of autonomous UAVs performing intelligence, surveillance and reconnaissance tasks. The focus is on coordinated target search (detection) and surveillance (monitoring) tasks. The developed technologies demonstrate how UAVs can plan missions collaboratively and re-plan adaptively based on real-time changes in UAV availability, pop-up targets and sensor capabilities. Metron has transitioned this UAV search technology to a NAVAIR Phase II SBIR contract to provide a new real-time search mission planning capability.

14. SUBJECT TERMS
Agents, Airlift, Unmanned Aerial Vehicles, Game Theory, Negotiation, Distributed Optimization, Search Theory, Bayesian Tracking

15. NUMBER OF PAGES
183

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This research was performed under contract F30602-00-C-0175 from the Defense Advanced Research Projects Agency (Taskable Agent Software Kit program). Technical direction was provided by Dr. James Hendler and Dr. Gary Koob. The author would also like to thank Mr. Stuart D. Draper at MITRE for his assistance in developing the military demand data sets used for the airlift experiments, and Dr. Tom Mifflin at Metron for introducing the concept and potential benefits of collaborative airlift planning to the author.

In addition, there were a number of mathematicians and computer scientists at Metron who developed the software testbeds and demonstrations, designed and conducted the experiments and performed the analyses presented in this final technical report. Dr. Chris Hellings designed the airlift infrastructure and developed the distributed auction mechanism described in this report. Mr. Aren Knutsen was responsible for the mission swapping mechanism and the airlift GUI used for demonstrations.

Mr. Knutsen and Mr. John Cunningham were primarily responsible for the UAV search and surveillance components, respectively, and shared the integration tasks on the joint search and surveillance effort. They were also the primary points of contact for several universities that used the Metron UAV simulator to aid their TASK research. Finally, Ms. Christine Judd extended the UAV search software to incorporate deployable unattended ground sensor networks, and Dr. Michael Greenblatt contributed to the development of the evasive target search algorithms.

1. INTRODUCTION

Current and future operations by the US military services will require intense collaboration within each service, across services, with other departments and agencies (e.g., State Department and CIA) and with our allies. Successful collaboration will also need to occur between the government and the private sector. Within the private sector, enterprises that normally compete with one another will have to cooperate to accomplish the goals of the operation. In fact, while the parties may agree on the basic operational goals, each party often will have its own sub-agenda and operating constraints. The collaborating parties also may not fully trust each other, and some may be in competition, economic or otherwise.

Under this DARPA TASK research contract, Metron has developed and implemented technology that addresses the dynamic problem of autonomous, competitive agents negotiating over the fair division of resources and tasks. In particular, we are interested in a better fundamental understanding of how to modify the rules of agent interactions to ensure that desirable system attributes, such as efficiency (no wasted utility) and stability (no incentive to cheat), are realized. Rosenschein and Zlotkin describe this type of design mechanism as "*social engineering for machines*" [RZ94].

Our research effort has three primary design themes: (1) procedures for fair division, (2) strategies that adapt based on historical agent interactions, and (3) negotiating protocols that ensure that the evolved strategies promote desirable system attributes. These research themes give us an opportunity to investigate the inverse problem of transforming a desired set of global attributes into an effective set of protocols that promote that behavior.

Our research addresses a wide class of large-scale, dynamic resource allocation problems. Traditional optimization approaches typically decompose large-scale, dynamic resource allocation problems into subproblems, each of which is optimized subject to a local resource budget assigned by the system. The process of determining the resource budget is called the “master problem”. Each subproblem yields a local solution—the union of which is the system solution—as well as a sensitivity analysis (called dual variables) that aids the master problem in modifying the local resource budget to improve the system solution. This process continues for many iterations using different local resource budgets.

These centralized approaches suffer from three primary weaknesses in practice: (1) the computation time can scale poorly with the number of iterations and subproblems, especially when the subproblems cannot be solved in parallel; (2) entities represented by a given subproblem (such as a commercial air carrier participating in a military airlift) may not want to reoptimize over multiple budget scenarios or share dual information that may aid one of its competitors in the next iteration; and (3) the final solution is fragile to uncertainty in the environment state, meaning that the entire optimization process may need to be repeated (often from scratch) as changes occur over time.

Addressing these weaknesses requires a radically different approach to solving the problem. Under this research effort, we focus on approaches to solving this class of dynamic resource allocation problems using a distributed, multi-agent framework. The key innovation is developing negotiation protocols (the public rules by which agents interact) that encourage autonomous agents performing local optimization to construct solutions that have desirable system attributes (e.g., efficiency, fairness, stability, simplicity, symmetry). This agent behavior is not forced or altruistic; rather the strategies that evolve or that the agent chooses—those strategies that maximize self-interest under a given set of negotiating protocols—also promote desirable system behavior.

We have incorporated these design elements into multi-agent systems in two different domains: (1) procuring commercial airlift to support strategic military airlifts and (2) coordinating a fleet of semi-autonomous, unmanned aerial vehicles (UAVs) performing intelligence, surveillance and reconnaissance (ISR) tasks on ground targets. The airlift problem is challenging because the natural competition between commercial air carriers means that cooperation cannot be guaranteed. In the UAV domain, the challenge is achieving real-time, distributed, effective coordination among a fleet of semi-autonomous UAVs.

For both domains, we perform extensive experiments to analyze the behavior of these multi-agent systems and validate the theoretical properties of these systems under different environmental settings and rules governing agent interaction. In the sections that follow, we introduce the various research elements in greater detail.

1.1. Motivating Application

We start with a motivating example that describes a seemingly plausible attempt by the FAA in the early 1990s to use collaboration for air traffic management that failed spectacularly. A close examination of the collaboration dynamics revealed the fatal flaws of the negotiation. Metron, under FAA R&D funding, was able to change the negotiation slightly by modifying the information shared and the incentives provided, which led to the creation of a successful system for both the air carriers and the FAA [Wam97, CHOSTW01].

1.1.1. Collaboration does not guarantee cooperation

The presence of collaboration among competing enterprises is not sufficient to ensure the cooperation necessary to satisfy collective goals. We present a real-world example in which the self-interests of the individuals dominate the overall behavior even though all parties agree that cooperation is the better solution.

The role of the FAA's Air Traffic Control System Command Center (ATCSCC) is to ensure that the aircraft flow from scheduled flights does not exceed the capacity at congested airports. In general, ATCSCC can accurately estimate capacity directly from weather reports and the towers of the affected airports. The actual airport demand is harder to estimate because each individual airline determines which flights it will fly or cancel on a given day.

Airlines pay a staggering cost due to poor traffic management. Underestimating demand causes planes to be delayed excessively in the air or diverted to other airports. Diversions can force passengers to be put up in a hotel overnight or crews and planes to end up in the wrong city. On the other hand, overestimating demand causes aircraft to be delayed unnecessarily on the ground while the supposedly congested airport had little incoming traffic.

The FAA proposed a direct solution called "Collaborative Traffic Flow Management." Each airline would provide real-time schedule data when congestion was expected due to bad weather. The FAA would then allocate arrival slots to scheduled flights and delete arrival slots from cancelled flights. FAA analyses showed this would lead to a near "optimal" traffic management solution. Furthermore, the airlines acknowledged that if every airline provided accurate data to the FAA, the resulting proposed solution would benefit all airlines. Nevertheless, not a single airline cooperated and the initial attempt at collaborative Traffic Management was a failure.

The proposed FAA scheme was seriously flawed. In particular, the FAA unintentionally penalized airlines that provided cancellation information during congested periods. The FAA assigns landing slots to each carrier in proportion to the number of scheduled arrival flights for that carrier. When a carrier cancelled a flight, that arrival slot was taken away from the carrier and given to another airline. Since carriers receive slots in proportion to arrival flights, the carrier with the cancellation was then doubly-penalized because the cancellation decreased the proportion of slots allocated to that carrier.

Consequently, one airline's cooperation benefited only its competitors. If all airlines had cooperated, then everyone would have benefited. However, if every airline but one had cooperated, then the renegade airline would benefit enormously without providing anything of value to its competitors. The scheme failed because it required the airlines to sacrifice self-interest for the "greater good" and left them vulnerable to cheating.

1.1.2. Metron's Collaborative Decision Making (CDM) program

After the original initiative failed, the FAA's R&D community tasked Metron's Aviation Division (which later became a separate company, Metron Aviation) to develop a prototype that would alleviate the problem. Metron created a system, called "schedule compression," that rewards cooperative behavior by the airlines [Wam97]. In particular, when an airline gives up an arrival slot that it cannot use, it is given the first available slot that it *can* use.

For example, suppose a United flight expected to arrive at 2:00pm is delayed until 3:00pm. Upon receiving this information, the FAA gives the 2:00pm slot to the first airline that can use it. Suppose that Delta has a 2:15pm arrival that can be moved up to 2:00pm. Delta benefits because its plane arrives 15 minutes earlier than scheduled, and United receives Delta's 2:15pm slot. Since the United flight will not arrive until 3:00pm, United rejects the slot, which then becomes available to the other airlines. If TWA takes that 2:15pm slot, then it gives its 2:30pm slot to United to accept or reject. This process continues until United receives a slot that it can use.

Under this scheme, all air carriers benefit, but the airline that benefits most is the one that donated the original slot. Instead of giving up a slot and getting nothing in return, the donating carrier gets a usable slot in the future. In addition, the rules reward airlines that provide up-to-date schedule information. *This new approach had the desired effect and was enthusiastically supported by the airlines.*

This research led to the development of the Flight Schedule Monitor (FSM) tool by Metron Aviation. FSM provides the FAA, NavCanada, and Collaborative Decision Making (CDM) participating airlines with the capability to monitor airport capacity/demand balance, model traffic flow management initiatives, and evaluate alternative approaches. FSM is also used by the Air Traffic Control System Command Center (ATCSCC) to implement Ground Stop (GS) and Ground Delay Program (GDP) strategies. Airline Operations Centers use FSM to assess the proposed GS/GDP, develop strategies to cope with the restrictions, and monitor GS/GDP initiatives that are in effect. FSM is used by more than 90 FAA facilities and 40 airlines in the United States and Canada.

The FAA and the air carriers have jointly invested over $25M into FSM, and in the first five years that FSM has been used operationally (since 2000), the carriers have measured a savings of nearly 30,000,000 delay minutes and $650M in direct operating costs and passenger and downstream delays using the FSM compression algorithms.

1.2. RESEARCH THEMES

In our research, we adopt, with slight modification, Wooldridge’s definition of an agent: “An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objective” [Woo97]. For our purposes, agents are autonomous entities that respond to their environment and typically interact with other agents in order to achieve their design goals. Depending on the domain, these interactions may require elements of cooperation, coordination, and negotiation. Furthermore, we assume that agents can adapt to their environment rather than merely respond.

Our research effort has three primary design themes: (1) procedures for fair division, (2) adaptive strategies based on the operating environment and historical agent interactions, and (3) negotiating protocols that ensure that the evolved strategies promote desirable system attributes. In addition, we will adopt technology from the multi-agent systems literature [Jen98, Jen00, Nwa96] and our own lessons learned from experiences with the FAA and commercial aviation community. We discuss each of these three research themes in some detail below.

1.2.1. Fair Division

The problem of agents sharing resources and dividing tasks has many practical applications. Brams and Taylor [BT96] wrote a book that collects procedures for doing fair division of goods and resolving disputes. Using these procedures, (human) agents can allocate assets fairly as part of a divorce settlement, negotiate new borders after a war, or divide chores that need to be performed. The emphasis is on providing allocations that are equitable, envy-free, and efficient (although it is difficult to achieve all three simultaneously when there are more than two agents).

The Brams and Taylor fair division procedures work best on problems with two agents interacting once. We describe below two procedures for two-agent interactions that create equitable allocations; one is also Pareto-efficient but vulnerable to deception (Adjusted Winner), and the other is immune to deception but not necessarily efficient (Proportional Allocation). When more than two agents interact, we will rely on the negotiating agent literature discussed later.

Consider an estate settlement with two heirs and two major assets, home equity and stock investments. Both assets have the same market value, but one heir prefers the home and the other prefers the stocks. That is, the perceived value that each heir places on each asset may be different from the market value. One equitable settlement would be to sell the home, and give half the proceeds (along with half of the stocks) to each heir. This settlement is also envy-free because neither heir would prefer the other’s allocation. However, it is not efficient in terms of Pareto-optimality because another allocation exists that one heir prefers without harming the other. A more efficient solution gives the house to the heir who preferred it and gives the stock to the other heir. This new allocation is equitable and envy-free as before, but it is also efficient.

The first two-agent procedure from Brams and Taylor is called Adjusted Winner (AW) in which *k* mostly-indivisible goods are divided between two agents. Under the AW procedure, one good (whose identity is not known before the negotiation) may have to be split. The AW procedure is envy-free (and consequently equitable since there are two agents) and efficient with respect to each agent’s *announced* preferences. Unfortunately, there is no incentive for the agents to announce truthful preferences. This can lead to one agent with complete information exploiting another that lacks information.

The AW procedure works as follows. Given k goods, G_1, G_2, …, G_k, let agent A announce points a_1, a_2, …, a_k for the k goods such that the points sum to 100. Let agent B do the same with announced points b_1, b_2, …, b_k. These points reflect the relative value placed on each good by each agent. Now, re-index the goods such that

$$a_1/b_1 \geq a_2/b_2 \geq \ldots \geq a_k/b_k.$$

Let r be the smallest index such that $a_1 + a_2 + \ldots + a_r \geq b_{r+1} + b_{r+2} + \ldots + b_k$. The AW solution gives goods 1 through r-1 to agent A and goods r+1 through k to agent B. Good r is divided between the two agents such that the two sums (representing the total perceived value of each agent's goods) are equal. Although the AW solution is envy-free and efficient, agents who are not truthful in announcing their points can manipulate it.

In the estate example, suppose the true valuation of the house and stocks is (60, 40) for agent A and (40, 60) for agent B. If they announce their true valuation, then A gets the house (60 points) and B gets the stocks (60 points). However, if A knows B's true valuation, then A can benefit by announcing a deceptive valuation of (50, 50). Agent A would get the house plus 1/11 of the stocks (54.5 points) and agent B would get 10/11 of the stocks (54.5 points). Although the announced points are the same, A receives goods worth 63.6 points of true value. Note that the deceptive solution (118.1 points) is also less efficient than the truthful solution (120 points).

The second two-agent procedure, called Proportional Allocation (PA), promotes truthfulness. The PA procedure is envy-free but not necessarily efficient, and it requires divisible goods. Given the set of announced points a_j and b_j, the PA solution gives agent A the fraction $a_j/(a_j + b_j)$ of good j and gives agent B the remainder. In the estate example, if the true valuations are announced, then A receives 60 percent (60/100) of the house and 40 percent (40/100) of the stocks (52 points) and B receives 40 percent (40/100) of the house and 60 percent (60/100) of the stocks (52 points).

If A uses the deceptive valuation, then A receives 55.6 percent (50/90) of the house and 45.5 percent (50/110) of the stock (50.5 points), and B receives 44.4 percent (40/90) of the house and 54.5 percent (60/110) of the stock (50.5 points). With respect to his true valuation, A would receive 51.5 points, which is worse than if he had told the truth. Note that deceptive and truthful solutions under PA (102 and 104 points, respectively) are less efficient than the AW solutions.

Although both procedures lead to envy-free solutions, the AW solution is guaranteed to be efficient as well. One way to implement a hybrid procedure is to use the AW solution unless one agent protests (suspects that he is being exploited), in which case the PA solution is used. Over repeated interactions, this hybrid procedure encourages truthful behavior from each agent.

1.2.2. Adaptive Strategies

The fair division procedures work best on problems with two agents interacting once. AW is efficient but vulnerable to deception, and PA is immune to deception but not necessarily efficient. Axelrod [Axe94] studied the two-person Iterated Prisoner's Dilemma in which cooperation rather than truthfulness was the encouraged trait. He showed that cooperation based upon reciprocity could evolve and sustain itself if the prospect of long-term interaction exists.

The difference between short-term and long-term interaction between agents is important. In the Prisoner's Dilemma payoff matrix shown in Table 1–1, the payoffs are such that the short-run optimal strategy for each agent is to defect. This is a dominant strategy—no matter what the second agent chooses to do, the first agent is better off defecting. Using this short-term strategy (Defect) over the long term hurts both agents. However, certain strategies can increase the long-term benefit of each agent. Strategies such as "Tit-for-Tat" (TFT), in which an agent cooperates unless its opponent defected on the previous move, can promote and reinforce cooperation.

(SENTENCE IN YEARS)	**AGENT 2**	
AGENT 1	**COOPERATE**	**DEFECT**
COOPERATE	(1,1)	(5,0)
DEFECT	(0,5)	(3,3)

Table 1–1: Sample Payoff Matrix for two-agent Prisoner's Dilemma

Strategies can also evolve automatically rather than through human invention. Genetic algorithms have been used successfully to find effective strategies in complex environments [Axe94, Axe97, Mat98, Ser96]. Axelrod found that the strategy of reciprocity or TFT, which had done well in direct competition with other strategies that people had devised, emerged from his evolutionary strategy experiments, thus validating the robustness of reciprocity.

The general approach for performing the genetic adaptation is as follows. A chromosome represents each strategy, and each gene in the chromosome represents the action that an agent would take under that strategy based on a particular state or history. The resulting chromosome contains the set of actions that would be taken under all possible states or histories under that strategy. Starting with an initial population of agents, each with a different (possibly random) chromosome, the agents interact and score points based on their actions. The simulation continues for a fixed number of interactions.

Chromosomes mate to create the next generation. The likelihood of a given chromosome mating is proportional to its score, so the next generation will receive more genetic material from the successful chromosomes than from unsuccessful ones. Given two chromosomes, crossover and mutation operations create two new offspring. The simulation continues for a fixed number of generations or until the population fitness score stabilizes.

Axelrod discovered several interesting phenomena after running these experiments [Axe97]. The first is the effect of noise; that is, misunderstanding or misapplying an action. If two agents using the TFT strategy interact repeatedly, then the expected payoff is high. However, if one agent defects accidentally, then a chain reaction of defections follows, alternating from one agent to the other. TFT is not robust to noise.

However, Axelrod found two attributes (generosity and contrition) that added robustness to his reciprocity strategies. *Generosity* means cooperating sometimes when the agent would otherwise defect. *Contrition* means not being provoked by an opponent’s response to an unintended defection. However, these concessions are not to be excessive—noise calls for forgiveness, but too much forgiveness invites exploitation.

Finally, Axelrod investigates promoting norms to create a self-policing system in which agents punish other agents who do not cooperate [Axe97]. *Norms* are how society describes acceptable behavior in a given setting. Agents that violate norms are often punished or ostracized. Existing norms can help explain whether cooperation succeeds or fails. Norms evolve in society. Consider how norms have changed in recent history regarding smoking in public or women working outside the home.

Another mechanism, called *metanorms*, helps norms emerge and prove stable. Metanorms reflect a willingness to punish violators of norms as well as those who fail to punish violators. Self-policing of norms and metanorms is essential in open, dynamic environments in which new agents enter the system and no central enforcement exists.

1.2.3. Negotiating Protocols

Rosenschein and Zlotkin [RZ94] point out that agents who can communicate and understand each other may not be able to come to agreements. *Protocols* are the public rules by which agents can come to agreements. These rules include the kinds of deals that can be made, the sequence of offers and counter-offers that are allowed, and the threats, promises and concessions that can be made.

A proper set of negotiating protocols, along with the requisite incentives and punishment mechanisms, can encourage individual designers to build a self-interested agent whose specific behavior also has desirable system attributes. This agent behavior is not forced or altruistic; rather it is that the strategies that the agent chooses or evolves—those that maximize self-interest under a particular set of negotiating protocols—also promote desirable system behavior.

Rosenschein and Zlotkin also describe a set of attributes that might be important to system designers:

- Efficiency – agents should not waste resources or utility when agreements are reached;

- Stability – agents should not have an incentive to deviate from agreed-upon strategies;

- Simplicity – interactions should involve minimal communication and resource demands;

- Distribution – interactions should not require a central decision-maker;

- Symmetry – no negotiating mechanism should treat agents differently due to inappropriate criteria (the appropriateness of the set of criteria may depend upon the domain).

Negotiating protocols do not need to include all attributes. In particular, the notion of stability may change meaning when each agent attempts to evolve its strategy to maximize self-interest. In that case, stability may be linked more closely to efficiency, in that strategies that evolve and increase system efficiency dominate those that decrease system efficiency.

Rosenschein and Zlotkin [RZ94] provide a broad review of game theoretic tools to guide the design of negotiating protocols. However, there are other references from which to draw. Binmore and Vulkan [BV97] apply game theory to autonomous agent negotiation as part of the Advanced Decision Environment for Process Tasks (ADEPT) project, which uses negotiating agents to provide quotes to design custom British Telecom networks for customers. Faratin *et al.* [FSJ98] build a mathematical model of contract scoring functions and define a negotiating thread to represent the sequence of offers and counter-offers between two negotiating agents.

Vulkan and Jennings [VJ00] modify English auction protocols to use in auctioning services. Two auction protocols rely on agents playing dominant strategies (strategies that yield higher expected payoffs regardless of other agents' behavior or state of the world). In the English auction protocol, an auctioneer raises the price until only one bidder remains. In the Vickrey auction [Vic61], which has simultaneous sealed bids, the highest bidder wins, but pays the second-highest bid amount. Although the seller receives less than the highest bid, the seller benefits because the Vickrey format encourages accurate bids. The winner pays less than his bid, and each bidder benefits from not wasting resources trying to outguess its opponents. This Vickrey format is used later in our collaborative airlift planning research.

Kraus and Lehmann developed an automated negotiating agent system that plays the board game Diplomacy [KL95]. Playing Diplomacy well requires a capacity to negotiate, explain, convince, promise, and keep or break promises. Kraus later investigated interdisciplinary approaches to negotiation [Kra97]. Finally, Matos *et al.* [MSJ98] developed a system in which successful negotiating strategies evolve using a genetic algorithm.

This game-theoretic approach to designing negotiating protocols assumes that agents act rationally. The assumption of rational behavior is fragile in the open market and can be dangerous economically to rational agents who interact with agents that have malicious intentions. This potential vulnerability also reinforces the need for a self-policing system that can identify and punish these destructive agents.

1.3. Collaborative Airlift Planning Overview

In the present day, there is greater military reliance on commercial assets and operational "best practices" of the commercial sector than in the past. This reliance will only increase in the future. In this section, we describe a domain area (commercial augmentation of military strategic airlift) that we believe is amenable to a multi-agent approach and supports the Department of Defense in making next-generation airlift procurement agreements more flexible.

1.3.1. Civil Reserve Air Fleet (CRAF) background

The air component of large military airlifts goes through the Air Mobility Command (AMC) based at Scott AFB. Under the Mobility Requirements Study 2005 (MRS-05), AMC uses commercial air carriers to airlift 93 percent of all troops and 41 percent of all long-range bulk air cargo through the Civil Reserve Air Fleet (CRAF) program. CRAF is a voluntary program in which commercial air carriers contractually agree to provide (for a fee) a fixed set of aircraft and crews to the military in times of need for a 45-day minimum. In return, participating carriers get the opportunity to bid on peacetime business.

Without this commercial airlift capacity, the military estimates that it would cost about $50 billion to procure and $3 billion per year to operate and maintain this airlift capacity as part of its organic fleet [Rob99]. However, there is significant cost to the military to guarantee this commercial airlift capacity. During peacetime, AMC spends about $650 million per year to charter commercial lift assets, partially as an insurance premium to the carriers to guarantee the needed airlift capacity in times of crisis.

During Desert Shield / Desert Storm, the airlift missions flown by the commercial carriers cost $2.3 billion, and for the period February to June 2003, the commercial airlift missions to support Operation Iraqi Freedom cost $1.2 billion [May03].

Although this peacetime business is attractive to many carriers, there can be a significant downside when the CRAF reserves are activated (such as during Desert Shield/Desert Storm) even though the military pays for the aircraft that are used. Some effects are short-term, such as having fewer aircraft available to satisfy the carrier's domestic schedule, and some are long-term, such as losing market share to a competitor who is not a CRAF participant. For example, during the first Gulf War, the CRAF fleet was activated during the peak holiday season in November-December 1990, which was extremely disruptive to the participating air carriers' domestic schedules.

Of particular concern is the inefficient way in which the military currently uses the commercial airlift capacity. AMC charters these commercial assets on a "mission by mission" basis, in which an aircraft and its crew are assigned a specific job. These assignments are not necessarily suited for the assigned carrier (in terms of proximity to available aircraft, for example), and carriers cannot request specific assignments (though they may volunteer in some cases).

This assignment approach ignores two particular carrier strengths, their command and control systems and their air operations personnel. Airlines have the tools and the people to solve large air operations problems. This includes the ability to create and analyze a concept of operations (such as a hub and spoke architecture), to plan flights and schedule crew and maintenance, and to leverage existing tools to execute the schedule smoothly. Not allowing the carriers to leverage these strengths increases the cost and reduces the flexibility in carrying out these missions.

After Desert Shield/Desert Storm revealed how disruptive CRAF activation could be for the air carriers, several carriers lost interest in the program. Given the voluntary nature of CRAF participation and the enormous cost to the military to acquire and maintain CRAF-equivalent airlift capability in its organic fleet, the military has needed to provide additional incentives (beyond eligibility for peacetime business) or higher rates in order to maintain adequate CRAF reserves.

1.3.2. Negotiation Protocols for strategic airlift

We believe that a multi-agent negotiation framework that allows the carriers to assert their interests as part of a collaborative airlift planning process will provide the necessary incentives to ensure future commercial carrier participation in CRAF. A multi-agent solution to this problem needs to satisfy the following properties:

- Allows commercial carriers to assert their private, competitive interests through negotiation agents rather than having to make those interests explicit and public;

- Provides incentives to carriers to volunteer assets early in the planning process;

- Enables the military to have its own agents that enforce airlift constraints such as delivery time windows and airfield congestion when evaluating offers from commercial agents;
- Enforces fairness in that no air carrier can be forced to provide more than its airlift obligation, but carriers who want additional business can bid on it; and
- Guarantees that negotiations continue and that the airlift assignment converges.

The competitive interests of the commercial carriers make this problem better suited for an agent-based approach than for classical optimization models. The agent approach allows a carrier to keep its enterprise rules private and to leverage its Air Operations Center expertise to improve its decision-making without making that expertise available to others.

We developed a multi-threaded, Java simulation for improving the strategic airlift procurement that lets the carriers negotiate their portion of the airlift rather than have the airlift divided and allocated in an arbitrary manner. Airlift missions are allocated to carriers using an auction plus swapping approach.

Inside the simulation, each carrier has an computerized bidding agent that computes a bid for each mission based on the carrier's cost structure, CRAF obligation and bidding strategy. If the reserve price set by AMC is satisfied, then the lowest bidder receives the mission and is paid the amount of the second-lowest bidder (Vickrey auction format [Vic61]). Otherwise, AMC assigns the mission to the carrier who has satisfied the least of its CRAF obligation. Furthermore, carriers can exchange missions with each other, as long as both parties agree.

Under this agent-based approach, the protocols create incentives to volunteer assets early in the planning process. Each carrier has a contractual CRAF obligation, and the amount that the military demands from that carrier is proportional to the size of the airlift. Once a carrier has fulfilled its fraction of the airlift voluntarily, it has no residual military obligation. Under this protocol, a carrier benefits from negotiating its airlift assignments early in the planning process, rather than waiting until the attractive movements have been satisfied by other carriers and having to fulfill its obligation with the remaining missions.

We have noted the advantages to the carriers, but there are benefits to the military as well. For example, the military has final control over the airlift assignments. When a carrier agent offers to satisfy an airlift requirement, a military agent can accept or reject that offer.

There are reasonable explanations for why an offer might be refused, such as an inappropriate aircraft type for the payload or for the runways at the arrival airfield.

These protocols leverage the fair division and negotiating protocol literature to ensure fairness in assignments and overall solution efficiency. Auction protocols can handle bids from multiple carriers for the same airlift requirement. To ensure accurate bids, the English or Vickrey auctions can be used depending on whether an open or sealed bidding environment is more appropriate.

Convergence is the most difficult property to ensure using a multi-agent system. For reasons of national security (and to deter wartime profiteering), the military can, at any time in the process, intervene and revert to the old CRAF style of allocating the airlift assignments. Under our proposed protocol, the unassigned movements would be assigned to each carrier according to its residual CRAF obligation. This provides further incentive for carriers to volunteer early for assignments to reduce their CRAF obligations and thus their vulnerability should CRAF be activated.

After developing the simulation, we conducted a series of experiments using a Desert Storm / Desert Shield-sized airlift scenario. The results show that this multi-agent auction plus swapping approach can cut in half the controllable operating cost and opportunity cost compared with the current centralized assignment procedure used today. This collaborative approach also makes plans more flexible, missions more reliable, and leverages commercial operational “best practices” without having to integrate those practices into military systems or to make the expertise publicly available to its commercial competitors.

1.4. UAV Coordination Overview

After completing our collaborative airlift planning research, we modified the agent protocols that we had developed to perform dynamic task allocation and negotiation for semi-autonomous UAV fleets coordinating ISR tasks. To evaluate these UAV protocols and distributed algorithms, we developed a Java-based simulation in which *M* UAVs with limited banking, sensing and communication capabilities focus on two types of ISR tasks: target search (detecting a set of stationary or mobile ground targets) and target surveillance (monitoring the locations of a set of mobile ground targets).

1.4.1. UAV Search (Target detection)

In the target search problem, UAVs collaboratively plan search paths to detect mobile targets whose locations are uncertain. We assume that the UAVs have an estimate of the target locations in the form of a spatial probability distribution, called the prior distribution on target location. Figure 1–1 shows two UAVs searching for a single target with a specified prior distribution. The cell color represents the probability of a target in that cell. The halo around each UAV is the sensor footprint, and the dots extending from each UAV show the negotiated search paths.

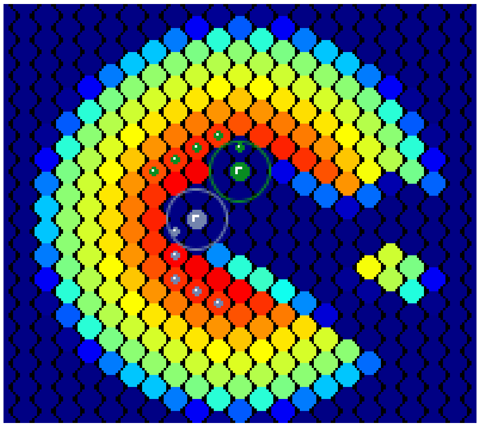


Figure 1–1: Illustration of two UAVs searching for a target

Each UAV optimizes its local search path by maximizing the expected number of targets detected over a finite-planning horizon, deconflicts with the search paths of the other UAVs to reduce duplicative coverage, and shares sensor reports with the other UAVs. We developed a genetic algorithm to cut down the combinatorial explosion associated with optimizing the search paths. In situations with limited bandwidth or communicates range, we developed an approach that we call delta synchronization to prioritize what information is shared between UAVs.

We use Bayesian likelihood functions and an estimated target motion model to fuse sensor information (which for our experiments is a simple, binary “detect” or “no detect” report) into the target prior distribution on location to produce a target posterior distribution. Likelihood functions provide a common currency for fusing information from different sensors. This Bayesian, nonlinear tracking approach easily incorporates non-Gaussian target priors, unlike linear Kalman filters, for example [SBC99].

1.4.2. UAV Surveillance (Target monitoring)

After being detected by a search UAV, a target is assigned to one of the surveillance UAVs. Each surveillance UAV has a set of targets for which it is responsible for maintaining target position estimates. The objective of the set of surveillance UAVs is to maintain tight position estimates on the set of moving targets over time, and the UAVs do this by visiting each target as frequently as possible. When a surveillance UAV passes over a target, the UAV sensor updates the target position estimate. To achieve this goal of visiting each target frequently, each UAV solves a Traveling Salesperson-type problem to decide in which order to visit its targets.

To improve surveillance, a UAV can propose three types of target trades with another UAV: (1) an even swap (exchange a pair of targets), (2) a pull (take a target from another UAV), or (3) a push (give a target to another UAV). The criteria for swapping (whether proposing or evaluating) may be greedy or cooperative and the amount of information shared by UAVs may be high or low. If the other UAV accepts the proposal, then the UAVs make the trade; otherwise, no trade occurs. These swap proposals and evaluations continue over time, with the UAVs taking turns proposing new swaps.

Figure 1–2 shows how trading targets leads to smaller UAV tours that eventually partition the space, with each UAV responsible for one sector. These sectors are not imposed, but rather they evolve naturally from the trading behavior of the locally optimizing UAVs. As the number of UAVs or targets changes, the UAVs can use these trading strategies to adapt their sectors quickly.

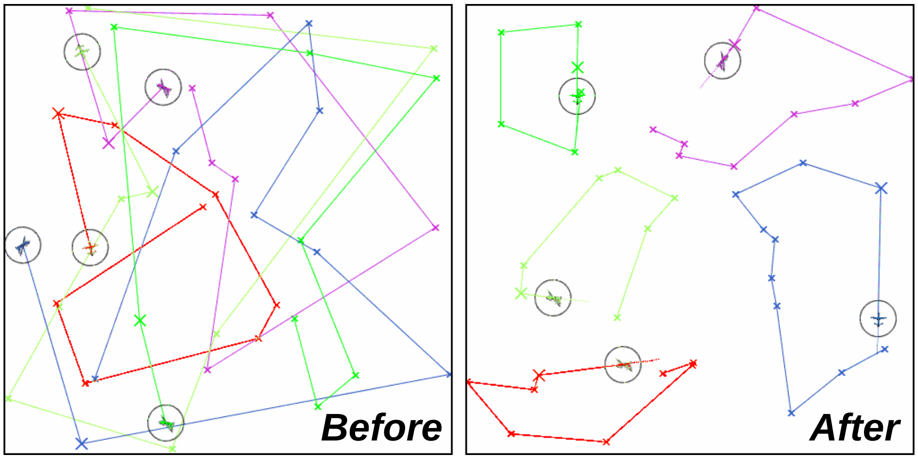


Figure 1–2: Effect of target swapping on UAV tour minimization

As targets move, a surveillance UAV will occasionally “lose” one of its targets. In this case, the surveillance UAV will make some effort to redetect the target by flying an ever-increasing spiral centered on the target's last known location. If the target still is not found, then the surveillance UAV passes the target information back to the set of search UAVs. The search UAVs fuse the information about when and where the target was last detected into its target probability maps with the goal of redetecting the target as quickly as possible.

1.4.3. Extensions to UAV Coordination

For one of our search extensions, we integrated sensor information from a network of Unattended Ground Sensors (UGS) into the target search problem. The UGS information fuses with the UAV sensor information via Bayesian likelihood functions directly into the target search probability maps. Consequently, the UAVs optimize their search paths with respect to UAV and UGS sensor information. We also developed a distributed, entropy-based strategy that enables each UAV to deploy a UGS node from a set of on-board sensors in order to resolve uncertainty regarding ground targets. These deployment decisions are made collaboratively across UAVs.

We also extended target search to consider evasive targets that have two motion components: a random element and an evasive element that depends on the locations and proximities of the UAVs. This change in the underlying target motion model changes how the UAVs update the evolution of the target probability maps over time. Our experiments show that modeling the evasive motion properly can increase the target detection rate by at least a factor of three.

Finally, we considered dynamic and autonomous self-organization of UAVs between target search and surveillance roles based on marginal value. The idea is that targets detected by the search UAVs are transferred to surveillance for monitoring. Over time, targets that get “lost” by the surveillance UAVs are transferred back to search for redetection. The research question that we address is whether UAVs can switch roles autonomously (with no outside direction) between search and surveillance based on the marginal value of each role. The experimental results show that the set of UAVs can switch roles effectively and efficiently in response to changes in the environment.

1.5. TECHNOLOGY TRANSITIONS

The military faces a multi-polar world in which there can be a threat anywhere, at any time, across the globe. These application areas and innovative multi-agent models can help lay the foundation for how the Department of Defense and other government and commercial enterprises interact in the future.

With the new search and surveillance capabilities that we have developed, UAVs can plan missions collaboratively and can re-plan adaptively based on real-time changes in UAV availability, pop-up targets and sensor capabilities. Metron has two official transitions of the UAV search technology, one to a DARPA DSO seedling contract and the other to Naval Air Systems Command (NAVAIR) Phase I and II SBIR contracts.

Under the DARPA DSO contract, Metron extended the UAV search capability in two fundamental ways. The first breakthrough is a value potential approach to optimizing search paths based on approximating an infinite-horizon search plan. Using this value potential to dictate UAV motion improves the search performance, especially for disjoint, multimodal ("patchy") probability distributions on target position.

The second innovation under the DSO work introduces dynamic area sectoring, which allows UAVs to partition the search area dynamically and to balance the search workload across UAVs. Sectoring also eliminates the need to deconflict search paths and simplifies collision avoidance because each UAV stays inside its sector. In our experimental testing, combining the value potential-based UAV motion and dynamic sectoring reduces the median time to target detection by up to forty percent compared with finite-horizon planning without dynamic sectoring.

For the NAVAIR SBIR contracts, Metron is developing a real-time, air mission planning component into the Undersea Warfare-Decision Support System (USW-DSS) program. The primary research and development efforts involve combining two Metron core technologies: (1) multi-sensor data fusion based on Likelihood Ratio Tracking (LRT) and (2) coordinated, real-time aircraft search based on distributed optimization. The aircraft search optimization component draws heavily on the research performed under this DARPA TASK contract and the DARPA DSO contract.

1.6. Report Outline

The remainder of this report is structured as follows. Chapter 2 describes the Virtual Transportation Company (VTC) concept. The VTC concept was designed to serve as a Research Exploration Framework (REF) for Metron and other contractors interested in this domain, including the University of Texas, Stanford University and Cornell University. The chapter provides an overview of the different types of scenario data sets that were developed for the researchers. We also provide details on the mission timing requirements and the economic cost and inventory models for carrier operations.

The various sections in Appendix A provide in-depth discussions of the scenario data sets, including specifications and examples of several of the data formats. The required data include movement requirement databases, aircraft planning factors, fleet information for CRAF participants, infrastructure such as airfield and runway information, and enterprise business rules for the commercial carriers.

Chapter 3 describes the details of our auction and swapping protocol approach to solving the collaborative airlift planning problem. We perform a series of experiments using the VTC scenarios to evaluate the performance of the multi-agent protocols. The first set of experiments compares the auction protocol with the assignment procedure that is currently in practice. The second set of experiments illustrates the effect of the auction reserve price on the negotiated allocation. The final set of experiments explores the computational effort associated with the protocols and investigates the consequences of unfair mission swapping.

Chapter 4 covers the target surveillance aspect of UAV coordination. We investigate both greedy and cooperative target swapping approaches with a series of experiments. The results show that high-quality system solutions can be obtained through local optimization by individual UAVs. In addition, we show how the rate of convergence to good system solutions can improve given cooperative UAV behavior (adherence to system goals rather than strictly local goals) and greater information sharing.

In Chapter 5, the focus changes to the target search aspect of UAV coordination. We describe a Bayesian likelihood approach to target search that relies on finite-horizon search path planning. To reduce the exponential explosion associated with the number of possible search paths, we develop a genetic algorithm to optimize the search paths. We perform a series of experiments that show the benefits of distributed, Bayesian search with respect to minimizing the median time to target detection and minimizing the average (or root-mean-squared) error associated with the estimated target location prior to detection.

Chapter 6 addresses three extensions to the basic search and surveillance technology. First, we integrate a network of unattended ground sensors (UGS) into the search problem, and demonstrate how UAVs can choose collaboratively when to deploy a UGS to minimize search effort. Second, we consider the effects of evasive targets that move partly in response to the UAV locations. Finally, we consider a joint search and surveillance problem. The surveillance UAVs maintain target positions while the search UAVs detect targets with unknown locations. The joint problem involves each UAV deciding whether to perform a search or surveillance role depending on the marginal value of each task at a given time. We show the value of our approach in a series of experiments over a wide range of environmental settings.

Finally, in Chapter 7, we describe our conclusions and two technology transitions that have resulted from this research. The first is a DARPA DSO seedling effort to improve the coordinated UAV search performance. The second transitions are NAVAIR SBIR Phase I and II contracts to prototype and develop a real-time, air mission planning component into the Undersea Warfare-Decision Support System (USW-DSS) program.

2. DESCRIPTION OF VIRTUAL TRANSPORTATION COMPANY CONCEPT

2.1. OBJECTIVE

In addition to performing research, our TASK contract specified that we were to develop a Research Exploration Framework (REF) related to the transportation logistics domain to be used by Metron and other interested TASK participants. The REF was designed to focus the research and to provide a common basis for comparing research results across groups. This framework was shaped by the following principles:

- The REF should be easily accessible to the researchers. While the nature of the problem should be complex, it should not require extensive independent effort on the part of the researchers to understand the domain or to acquire the necessary databases, operating parameters, etc. Ideally, the information in this chapter and the electronic versions of the appendices would be sufficient for all of the researchers that selected this REF.

- The REF should be difficult, perhaps even impossible, to solve by traditional methods. Making significant progress on this problem should require the development of new mathematical and computer science techniques.

- The REF should be a good candidate for collaborative, distributed systems technology. A problem that begs a solution through centralized computing and a tight control structure is not a good candidate for this program.

- The REF should have relevance to the Department of Defense (DOD).

The REF we chose is the problem of leveraging commercial transportation assets for military use in times of crisis in a mutually beneficial manner, which we called the "Virtual

Transportation Company" (VTC). Although the domain was logistics related, the research goal was to develop a better understanding of the general problem of dynamically acquiring resources to satisfy tasks in which the resource owners may be competitive and non-cooperative.

The information in this chapter was drawn from a REF white paper written by Metron and distributed to the TASK researchers early in the program [GM01]. We describe the military strategic lift problem and supporting data sets used to perform experiments. However, the goal of the research was to focus on techniques that apply to the more general acquisition problem. We understood that researchers should not be required to become logistics experts in order to apply their technology to this class of problems. Consequently, we introduced simplifications that distilled the essential elements of the problem for the researchers. In addition, we identified opportunities for researchers to add or subtract detail depending on their interests.

In addition to Metron, four other research groups participated in the Airlift REF: Stanford University, Cornell University / University of Washington, University of Michigan and University of Texas at Austin. The high-level research questions addressed by these groups are as follows:

- What effect do individual agent strategies and fairness criteria have on solution quality, convergence, and other properties of the final solution? (Stanford, Metron)
- What impact does this structure have on the effective complexity of the approach? (Cornell/UWash)
- How can we achieve solution robustness as commitments change in near real time? (Michigan)
- How can a solution containing the elements above be practically designed and implemented? (Texas, Metron)

A few months after 11 September 2001, the TASK program shifted focus away from the VTC REF and toward the UAV coordination domain discussed later in this report. Consequently, the VTC REF was retired prematurely, with only a few of the open research issues resolved. At the end of this chapter, we highlight some results for each of the research groups, and in Chapter 3, we describes Metron's approach and results in greater detail.

2.2. Overview of VTC Problem Structure

In addition to being called the VTC REF, there was also a longer, more technical name for the REF that reflected the more general problem space addressed by the technologies: "Large-scale, Collaborative, Dynamic Resource Allocation among Competing Enterprises" (LCD RACE). Briefly, the VTC problem investigates how DOD could use transportation assets more effectively to perform a strategic lift during a major contingency? The assets may be owned by DOD (*organic* assets) or temporarily acquired from the commercial sector.

The DOD has an existing process for moving equipment and people into, and out of, the theater of operations (see Figure 2–1). It is characterized by centralized planning, in the form of deterministic scheduling, and re-planning in reaction to real-time events. In the planning stage, the military identifies individual missions, which are then broken off and assigned to an organic asset (e.g., a C17) or contracted out to the commercial sector (e.g., a United Airlines 747). The commercial assets used by DOD consist of platforms (planes, ships, trucks, etc.) and crews. Current practices do not leverage off the considerable information systems resident at the enterprises. Consequently, DOD cannot operate the combined asset fleet as efficiently as a commercial enterprise, such as Federal Express.

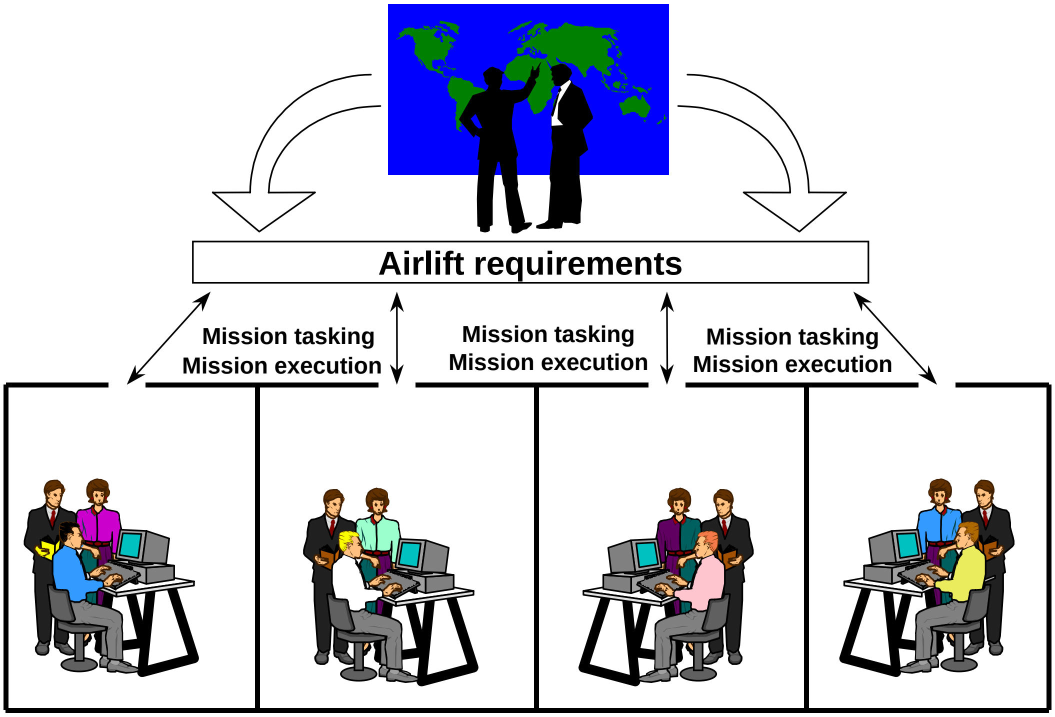


Figure 2–1: Current way in which the military uses commercial transportation assets

Furthermore, the current way of doing business is reactive. While there is widespread recognition that strategic lift possesses inherent uncertainties, there is little attempt to model those uncertainties, much less to optimize across them. Figure 2–2 illustrates a collaborative approach for solving the airlift portion of the VTC problem. Under this new approach, the commercial enterprises work with the military to provide sufficient lift in a manner that increases the flexibility and reliability of the missions for the military, and reduces the cost and disruption for the commercial enterprises.

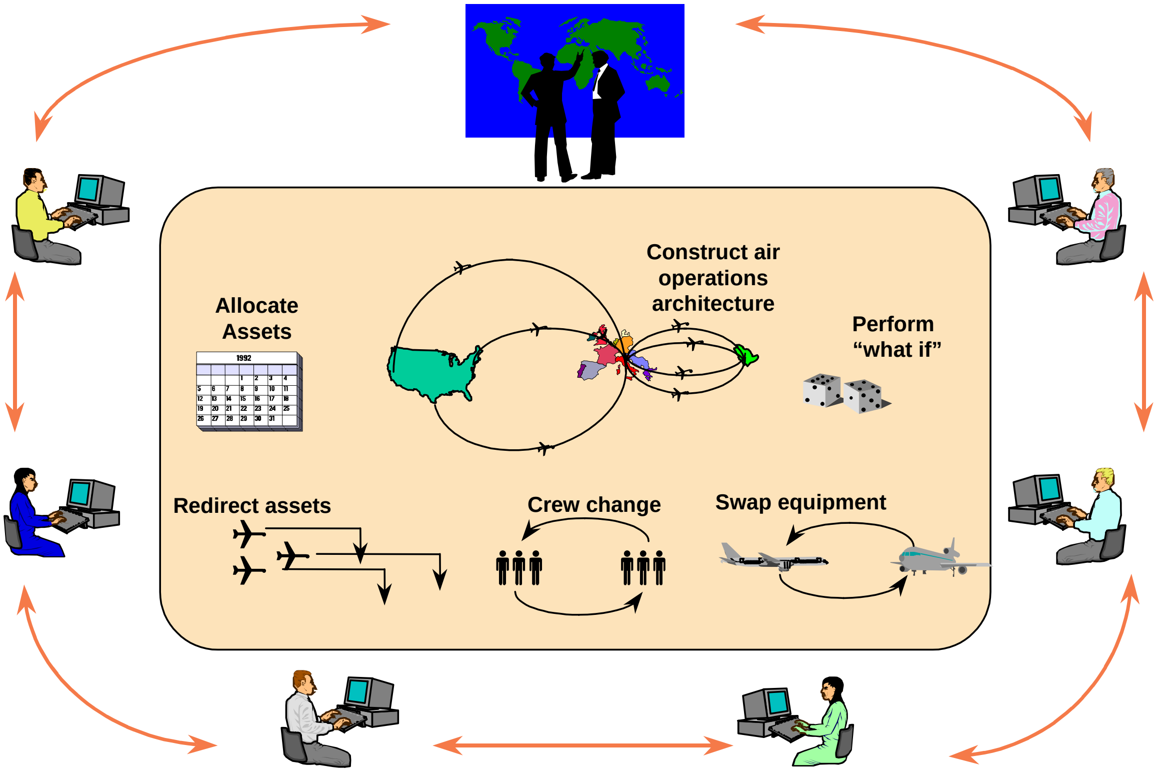


Figure 2–2: Collaborative approach to the Virtual Transportation Company problem

The VTC problem structure can be partitioned into the following categories: Enterprises, Demand, Infrastructure, and Regulations and Contracts. We present a brief summary of each category below and further details are provided in later sections and in Appendix A. The VTC problem can be stated in general as follows: How can the enterprises satisfy dynamic demand requirements at minimal cost without violating constraints imposed by the infrastructure, regulations and contracts?

While the VTC is certainly a logistics problem, the statement above could easily apply to non-logistics applications. For example, consider a phone system that automatically negotiates real-time rates for long-distance calls from multiple telecommunications providers. The technology required to solve the VTC could be transitioned to this new domain with minimal changes. In fact, we leveraged the mission swapping protocol described in Chapter 3 to prototype quickly a UAV surveillance demonstration for the TASK program office when the UAV domain was being considered for adoption. As we describe the VTC in its natural logistics setting, we will illustrate how the different elements can generalize, when possible, to more diverse domains outside of logistics.

2.3. VTC Elements

2.3.1. Enterprises

Enterprises include all affected organizations, both commercial and DOD, such as the United States Transportation Command (USTRANSCOM), American Airlines, etc., along with their assets and business models. In reality, many organizations play a role in strategic lift. For the VTC REF, we restrict the problem to two government enterprises, USTRANSCOM and the Commander-in-Chief (CINC). In our formulation, the CINC determines contingency demand and sets priorities, both initially and throughout the contingency. In practice, demand may be shaped by the individual services, Congress, the Office of the President, the Joint Staff, etc.

Likewise, we assume that USTRANSCOM is solely responsible for satisfying the demand by allocating, procuring and scheduling transportation assets. For our purposes, we treat entities such as the Air Mobility Command and Military Traffic Management Command as part of a monolithic USTRANSCOM.

Both the military and commercial enterprises (including air carriers, trucking, rail and shipping companies) supply transportation assets. Appendix A.3 lists the different types of transportation assets, along with characteristics such as capacity, speed, mode and maximum range. To reduce detail and overhead, we have represented only a subset of all asset types. The format for describing the transportation fleet of each enterprise (one military and the rest commercial) appears in Appendix A.5. The economic model used by each enterprise to compute the cost of using a specific asset at a specific time appears in Section 2.7 and Appendix A.6.

2.3.2. Demand

The demand for transportation assets divides into contingency demand and all other demand, including commercial demand. We represent the contingency demand with a modified version of a Time-Phased Force Deployment Database (TPFDD). A TPFDD is a list of individual movement requirements stating that quantity *w* needs to move from location *x* to location *y* by time *z*. The TPFDD we will use is based on an airlift scenario developed for DARPA by USTRANSCOM.

We have modified the TPFDD in two ways. First, we have reduced the number of data fields used to describe each movement requirement to its essentials. Second, we created new data fields that represent future events. For example, each line item lists the estimated number of passengers and tons of cargo for that movement. We have added another set of passenger and cargo data that represents the actual number of passengers and cargo for that movement, as well as a field that identifies the day on which the updated information becomes known. The database fields are described further in Section 2.6 and Appendix A.2.

Commercial demand will be handled differently. Each enterprise has a model of its daily operations. Assets provided to the military are assets that cannot be used to satisfy the enterprise's domestic schedule. Consequently, each time the enterprise provides an asset to the military, the enterprise incurs an opportunity cost (in terms of commercial business) that may not be offset by profit on the military mission. As more assets are provided, the opportunity cost increases due to extra delays and cancellations that result.

The commercial demand that we construct is used to track the fleet of available resources for each carrier and to provide an opportunity cost model for providing an asset to the military for a given amount of time. Rather than provide the volumes of data necessary to derive the individual opportunity cost functions, we provide instead the opportunity cost functions directly. This is another instance of isolating the logistics details from the multi-agent research whenever possible. Additional details on the enterprise models can be found in Section 2.7 and Appendix A.6.

2.3.3. Infrastructure

Infrastructure describes characteristics of roads, rail lines, rail yards, airfields, ports, etc. Of particular interest in logistics is the throughput capacity at the consolidation points. A successful solution of the VTC problem should be robust. This means that the solution has some degree of flexibility for handling problems such as equipment breaking at a rail yard or a weather front slowing the number of arrivals into an airfield. Some smaller foreign airfields, for example, may be overwhelmed by the arrival demand of a military airlift. In that case, the flow of people and goods into the airfield must be smoothed out as much as possible.

As we observed with modeling commercial demand, a researcher may choose not to include infrastructure constraints if it adds an unnecessary level of detail to their formulation. Appendix A.4 lists the field descriptions of the locations used in this test bed and a great-circle distance formula that can be used to compute distances between two points on the earth.

2.3.4. Regulations and Contracts – the Civil Reserve Air Fleet (CRAF)

Due to the economic impact of losing assets used in their daily operations, the commercial sector may be unwilling to supply assets at DOD rates during the contingency. In that case, there is a mechanism for the military to temporarily acquire the air and sea assets through the Civil Reserve Air Fleet (CRAF) agreement and the Voluntary Intermodal Sealift Agreement (VISA) program, respectively.

These arrangements are important for the DOD. For example, DOD plans call for commercial air carriers to airlift 93 percent of all soldiers and 41 percent of all airlifted cargo during crises. This airlift capacity would cost the military $50B to procure and about $3B per year to operate and maintain as part of its organic fleet [Rob99]. The military created the CRAF program to support these airlifts. CRAF is a voluntary program in which commercial air carriers contractually agree to provide a fixed set of aircraft and crews to the military in times of crisis in return for the opportunity to bid on peacetime business. Essentially, CRAF is an insurance policy for the military in which the peacetime premiums paid to the carriers guarantee the availability of airlift capacity during crises.

Although this peacetime business is attractive to many carriers, activation of the CRAF fleet (such as during the 1990-91 Gulf War) can be extremely disruptive to the air carriers' domestic schedules, especially during peak holiday seasons. Some effects are short-term, such as having fewer aircraft available to satisfy the carrier's domestic schedule, and some are long-term, such as losing market share to a competitor who is not a CRAF participant. In fact, some airlines have concluded that the potential peacetime business is not worth the risk of CRAF activation and have dropped out. Consequently, the premiums paid for this airlift capacity have increased significantly over the past fifteen years.

One reason why CRAF activation is so disruptive to the carriers is the "mission-by-mission" basis by which it is assigned tasking. The air component of large military airlifts goes through the Air Mobility Command (AMC) based at Scott Air Force Base. In the planning stage, AMC identifies individual missions that are then assigned to an organic asset (e.g., a C17) or contracted out to the commercial sector (e.g., a United Airlines 747). These assignments are made according to CRAF obligation (or volunteered assets) without accounting for carrier preferences (such as proximity to a hub or available aircraft). Using the carriers in this manner fails to take advantage of their primary strengths – their command and control systems and air operations personnel. Carriers have the tools and personnel to schedule and execute large, dynamic air operations, but they lack the autonomy and flexibility to manage their share of the airlift more efficiently.

An important element of these programs is that volunteered assets count against a carrier's CRAF or VISA obligation. In other words, suppose that a carrier volunteers assets to satisfy a subset of the movement requirements. If CRAF or VISA is activated to raise additional assets, then the carrier's remaining contractual obligation is reduced by the amount of lift that they already volunteered. This becomes important if CRAF or VISA is activated after some carriers have fulfilled their share of the lift and some carriers have not. Those that did not provide sufficient lift will be given the burden of satisfying the unassigned movements. By volunteering, not only do the carriers fulfill their obligation, but they can also select preferable movements to satisfy rather than getting what is left. Additional details on these programs are given in Appendix A.7.

An important aspect to consider is how to model the tradeoff between short-term and long-term profits for these enterprises. Suppose that a carrier signs up for the CRAF or VISA program in order to become eligible for peacetime business. If a military contingency arises in which the program is activated, then the disruption to the carrier's operations, not to mention its bottom line, might be greater than expected. If a carrier perceives that the risk of future activation imposes too high of a cost, then it may choose to drop out of the program.

This represents a danger to DOD because they rely on the availability of this commercial lift capacity. Consequently, one of the interesting aspects of this VTC problem is the economic equilibrium that is desired. Casual activation of CRAF or VISA may provide the lift assets that DOD requires in the short-term, but carriers may become less likely to participate in the future. Furthermore, if the carriers can provide their required lift assets at lower costs using the proposed collaborative VTC approach, then DOD will not need to activate CRAF or VISA as often nor increase the rates paid to the carriers to ensure future participation. Designing the interaction protocols among enterprises that provide these efficiencies is a critical aspect of this research.

The CRAF and VISA programs used today are devices to ensure cooperation from the commercial carriers. However, developing alternatives to these programs is one of the desired research topics in this program. In other words, how can cooperation be encouraged among the enterprises? How can the military be assured that sufficient assets will be provided at a reasonable cost in times of war?

2.4. Role of Multi-Agent Systems

We believe that the VTC is a good candidate for multi-agent system (MAS) technology. For example, the stakeholders and decision-makers are not, in general, co-located. The participating enterprises do not share the same goals or priorities. Some of them are economic competitors and insist on keeping their cost and revenue models private. Each is capable of autonomous action.

Agents could represent the interests of the participating enterprises, both DOD and commercial. One research issue to consider is the interaction between the agents representing the government and the commercial enterprises. The DOD has no preference regarding which enterprises should transport which movements (aside from equipment compatibility). They are concerned only that each movement is assigned and delivered on time. If no enterprise volunteers for a particular movement, then the DOD may choose to use organic assets or raise the incentives for the commercial agents. Pricing models are crucial in these negotiations. In order to have a common basis for making decisions, all relevant factors must be converted into monetary terms.

2.5. Solution Characteristics

A feasible solution to the VTC problem assigns assets to all of the military movement requirements subject to the timing and infrastructure constraints. However, different solutions may have different impacts on the participants. For example, activating CRAF and VISA leads to feasible solutions, but these solutions are typically expensive for the carriers in terms of opportunity cost and the arbitrary assignment of missions.

In the fair division literature [BT96], allocations can have characteristics such as efficient, proportional, and envy-free. In an *efficient* (or Pareto-optimal) solution, a carrier cannot improve its lift assignments without negatively impacting another carrier. For example, two carriers may swap a few assignments and achieve a mutual benefit because the new assignments are closer to their respective hubs.

However, efficient solutions are not necessarily good (fair) solutions. For example, a solution in which a single carrier is forced to provide all of the lift capacity and no other carriers want to participate is a bad solution. However, this solution is efficient because the forced carrier cannot improve its position without negatively impacting the other carriers.

A *proportional* solution has each carrier assigned to no more than its obligation, unless it chooses to volunteer additional assets. If at least one carrier volunteers more than its share, then the assets required from the other carriers should be less than the original obligation.

For carriers with equal obligations, an *envy-free* solution means each carrier prefers its set of movement assignments (in terms of operating profit) to the set of movements assigned to any other carrier. In other words, each carrier does not envy or prefer the assignments of any other carrier. Envy-freeness can also be extended to carriers with unequal obligations.

Envy-freeness is often incompatible with efficiency. Take a solution that is originally envy-free but not efficient. Suppose a single swap of assignments makes the solution efficient in that the two carriers involved benefit and all other carriers stay the same. This new solution may no longer be envy-free because one of the carriers whose assignment did not change may prefer the set of assignments of a carrier who did swap. In general, there is no “best” set of characteristics that a solution should have, but having defined these multiple characteristics allows us to describe the different types of solutions more fully.

2.6. MILITARY DEMAND (BASED ON A TPFDD)

In this section, we describe the VTC problem in more precise detail. As mentioned previously, much of this detail is specific to the logistics elements of the problem. The goal of the TASK research is to define the laws that govern distributed collaborative systems, not to extend the state-of-the-art in logistics.

In fact, by introducing additional levels of logistics details, we can make the VTC much harder to solve (and from a software development perspective, harder to implement and maintain) without getting any closer to the primary goals of the research. Consequently, we introduce the logistics details for convenience and explain how many of these details can be eliminated without losing the essence of what makes this an interesting (and extensible) problem.

We will use the following notation to describe the demand. We start with a deterministic description, and then add stochastic elements to add realism and difficulty to the problem.

2.6.1. Deterministic Description

Let $M = \{1, 2, \ldots, m\}$ be the set of all movement requirements. Associated with each movement requirement $m \in M$, we define

w_m = the payload vector (passengers, bulk, oversize, outsize) for movement *m*.

Passengers are measured by the number of personnel to be transported. Bulk, oversize, and outsize are different types of cargo, measured in short tons. Due to commercial aircraft configurations (such as door size and floor strength), we assume that commercial air carriers can move only bulk cargo. Sea assets and military aircraft can move any cargo type. We can relax and aggregate the cargo assumptions without loss of generality (at the expense of losing realism).

Each movement requirement has three legs: origin to point of embarkation (POE), POE to point of debarkation (POD), and POD to destination. Each leg has a required mode of transportation, either land, sea or air. We define

J = the set of locations in the problem;

K = {“land,” “sea,” “air”}, the set of all modes;

L = {1, 2, 3}, the set of legs for each movement.

Associated with leg $l \in L$ of movement $m \in M$, we define

$(i_{lm}, j_{lm}, k_{lm}) \in J \times J \times K$, the origin, destination and mode for movement *m*, leg *l*.

This is not the most compact representation because the end location of one leg is the starting location of the next leg. That is, for each movement requirement, $j_{1m} = i_{2m}$ and $j_{2m} = i_{3m}$, representing the common POE and POD locations, respectively. However, this expanded form allows each leg to be represented independent of the others.

There are two ways to simplify the logistics detail associated with the legs. First, one could focus exclusively on the second leg (POE to POD), which is typically the long-haul leg. Doing so reduces the intermodal aspects of the problem. By intermodal, we mean that each movement requires multiple legs and (possibly) multiple assets that must be coordinated in time (the truck delivers the goods to an airplane that flies across the ocean to an awaiting railcar). By focusing only on the second leg, coordinating and negotiating multiple assets disappears.

The second simplification is to consider only the airlift portion of the long-haul legs. Doing so reduces the data overhead required to maintain multiple types of commercial enterprises. For the experiments that we performed and that are presented in Chapter 3, we considered only the long-haul airlift missions.

Having discussed the locations and payload, we add the third element of each movement, the timing. We specify a Ready-to-Load date (RLD), representing the earliest date that the payload can be loaded at the movement origin (i_{1m}), and a Required Delivery date (RDD), representing the latest date that the payload can arrive at the movement destination (j_{3m}). Given this movement time window and estimates of the expected travel time on each leg by the specified mode (k_{lm}), a delivery time window can be computed for each leg $l \in L$ of movement $m \in M$,

t_{lm} = the expected travel time for movement *m*, leg *l* (specific to locations and mode);

u_{lm} = the earliest arrival date (EAD) for movement *m*, leg *l*;

v_{lm} = the latest arrival date (LAD) for movement *m*, leg *l*.

The delivery time window for each leg is not fixed because it is affected by the delivery dates of the other legs. For example, Figure 2–3 illustrates the delivery time window on each leg of a movement with RLD=3, RDD=11, and t_m = (1,2,1). In this case, we can compute initial time windows on the three legs of [4,8], [6,10], and [7,11]. If the first leg is completed on day 7, then the required travel time causes the time windows of the later two legs to shrink to [9,10] and [10,11], respectively.

Similarly, if the third leg is scheduled for delivery on day 7, then the feasible delivery windows on the first two legs shrink to [4,4] and [6,6], respectively. In all cases, however, there are two time window components that are fixed, u_{1m} = RLD$_m$ + t_{1m} and v_{3m} = RDD$_m$. The other components can float based on the scheduled delivery date of each leg.

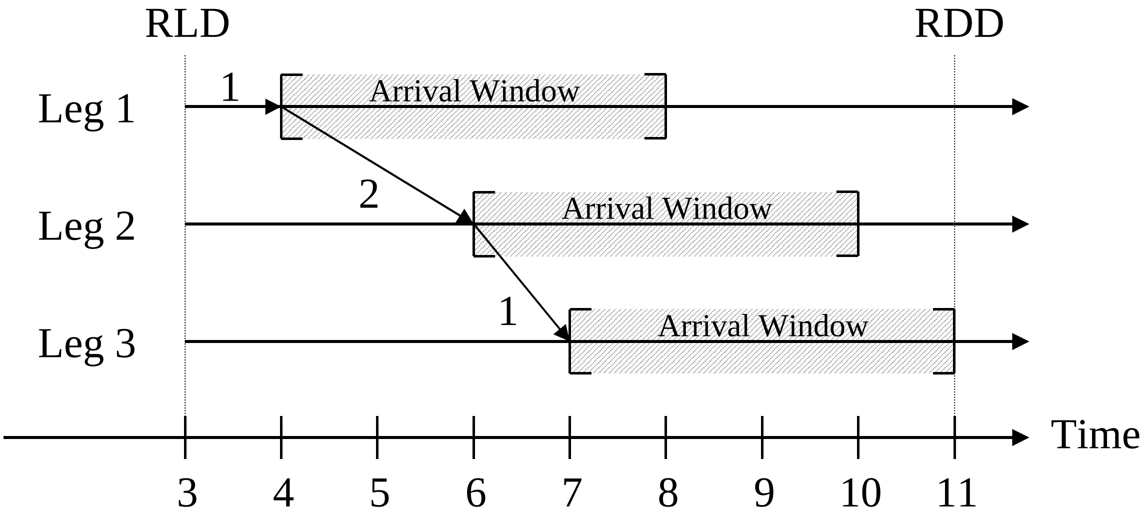


Figure 2–3: Illustration of arrival (or delivery) time windows on individual movement legs

Coordinating the delivery of the three movement legs subject to the time window constraints is a difficult part of the VTC problem. In fact, before the VTC REF ended, the University of Michigan had planned to feature this aspect of the problem in their commitment management research. There are, however, useful extensions of the VTC into non-logistics domains that do not require this type of coordination. Consequently, we consider the timing constraints alone to be essential to the VTC problem, but not the coordination of the timing of the three transport legs (which may be more useful in a logistics setting). For this reason (and to simplify the setting), we include explicitly the EAD and LAD for the second leg (long-haul leg) of each movement in the data sets.

2.6.2. Stochastic Components

Thus far, we have assumed that the TPFDD information is deterministic. In practice, this is an unreasonable assumption. Over the course of a strategic lift, priorities change, payloads change, and movements can be added or deleted. In order to model this uncertainty in a way that minimizes data requirements, while enabling reproducible simulation results (if desired), we developed a multiple-scenario approach.

Associated with each movement requirement *m*, there are two estimates of the payload vector, w_m^1 and w_m^2, the locations and modes, $(i_{lm}^1, j_{lm}^1, k_{lm}^1)$ and $(i_{lm}^2, j_{lm}^2, k_{lm}^2)$, and time window constraints, $(\text{RLD}_m^1, \text{RDD}_m^1)$ and $(\text{RLD}_m^2, \text{RDD}_m^2)$. In addition, there are random variables, s_m^1 and s_m^2, representing the times at which each scenario is "announced." To ensure that the scenarios are announced at least two days before the Ready-to-Load data, let

$$s_m^2 \sim U\left[0, \max\left(0, \min\left(\text{RLD}_m^1, \text{RLD}_m^2\right) - 2\right)\right]$$

$$s_m^1 \sim U\left[0, s_m^2\right]$$

That is, the switch between scenarios 1 and 2 is uniformly distributed between day 0 and two days before the earlier of the two RLDs. If the latter condition is prior to day 0, then set $s_m = 0$.

The TPFDD contains random samples $s_m^1(\omega)$ and $s_m^2(\omega)$ drawn from these distributions. Given these samples, estimates of the payload and timing constraints on day *t* are

$$\hat{w}_m(t) = \begin{cases} w_m^1 & \text{if } s_m^1(\omega) \le t < s_m^2(\omega) \\ w_m^2 & \text{if } t \ge s_m^2(\omega) \end{cases} \tag{2.1}$$

$$\text{R}\hat{\text{L}}\text{D}_m(t) = \begin{cases} \text{RLD}_m^1 & \text{if } s_m^1(\omega) \le t < s_m^2(\omega) \\ \text{RLD}_m^2 & \text{if } t \ge s_m^2(\omega) \end{cases} \tag{2.2}$$

$$\text{R}\hat{\text{D}}\text{D}_m(t) = \begin{cases} \text{RDD}_m^1 & \text{if } s_m^1(\omega) \le t < s_m^2(\omega) \\ \text{RDD}_m^2 & \text{if } t \ge s_m^2(\omega) \end{cases} \tag{2.3}$$

The locations and modes for the movement legs can be defined similarly. This representation can model a wide range of lift uncertainty. Aside from the obvious changes to payload and timing, movements can also be added or deleted this way. If $w_m^1 = (0, 0, 0, 0)$ and

w_m^2 is non-zero, then this movement can be considered as “added” at time $s_m^2(\omega)$. Similarly, if w_m^1 is non-zero and $w_m^2 = (0,0,0,0)$, then this movement can be considered as “deleted” at time $s_m^2(\omega)$.

Of course, this formulation is effective only if enterprises do not “anticipate” the future by cheating and looking ahead in time. For a subtle example of cheating, consider an added movement. Although the payload is initially zero, an enterprise could infer that a non-zero payload will be added in the future with estimates of the locations and modes. The enterprise decision-support function should be designed to ignore this type of anticipated information.

This randomness may affect an enterprise’s decisions. For example, enterprises that volunteer assets in the early stages of the contingency may regret their decision if new movement requirements, perhaps better suited to their operations, arise later. Similarly, enterprises that wait for better movement requirements may be stuck with the movements that the other carriers turned down. Furthermore, enterprises may select movements that vanish or are needed on a different day.

2.7. Enterprise Supply Model

Each enterprise (commercial and military) has a fleet from which to draw transportation assets for strategic lift. These assets are of four general classes: aircraft, ships, trains and trucks. Each class can be further subdivided into asset types with similar characteristics. In particular, let

B = the set of all enterprises (commercial and military);

A = the set of all asset types;

$A_b \subset A$; the set of asset types owned by enterprise $b \in B$.

Note that the commercial and military enterprises are part of the same group. In fact, the military and commercial enterprises are similar. They both provide assets for the strategic lift and both want to minimize their cost to perform their portion of the lift. The primary differences are that the military enterprise has no strict obligation to provide lift and has no other exogenous demand. In this respect, the military enterprise can be thought of as a special type of commercial enterprise.

Associated with each asset type is a set of performance characteristics such as speed, range, capacity, and transportation mode. We focus on the last two attributes and express them as

w^a = the payload capacity vector for asset type $a \in A$;

$k^a \in K$; transportation mode associated with asset type $a \in A$.

To reduce bookkeeping requirements, we are using only a subset of the asset types found in practice. The selected asset types and characteristics for organic and commercial assets can be found in Appendix A.3.

To define the total fleet inventory for each enterprise, let

D_{ba} = the number of assets of type $a \in A_b$ owned by enterprise $b \in B$.

The quantity D_{ba} is the total fleet inventory; the entire fleet is not necessarily available at any given moment because of the enterprise's daily operations. If there are assets that are not fully utilized, then it is in the enterprise's interest to volunteer the assets if there is a profit potential. On the other hand, if a enterprise is "forced to volunteer" assets due to threat of CRAF or VISA activation, then the enterprise must estimate the expected lost opportunity cost of the volunteered assets. Delays and cancellations caused by a reduced fleet add to this lost opportunity cost. By measuring this cost, enterprises can select movements that maximize profits.

Fundamentally, the enterprise must answer the question: What is the opportunity cost (in terms of its daily operation) of providing to the military an asset of type $a \in A_b$ leaving location $i \in I$ at time t_0 and returning to location $j \in J$ at time t_1? To answer this question, we build a model of the enterprise's daily operations. After building this model, we show how one could substitute a relatively simple cost function to approximate the output of the cost model.

We assume that each enterprise's fleet is spread out over a relatively small number of hub locations. Each day, we assume that assets leave the hub at a given time, are unavailable for some time, and then return to the hub. This out-and-back shuttle may repeat several times during the day. We also assume that the assets earn revenue and accumulate cost in rough proportion to the asset payload capacity and the amount of time that the asset spends away from the hub. Further details on the economic model are described in Appendix A.6.

At any given time, we can estimate the number of assets that sit at the hub awaiting work. Figure 2–4 illustrates the available assets for a single enterprise at a single hub. The available inventory decreases by one for each departure and increases by one for each arrival.

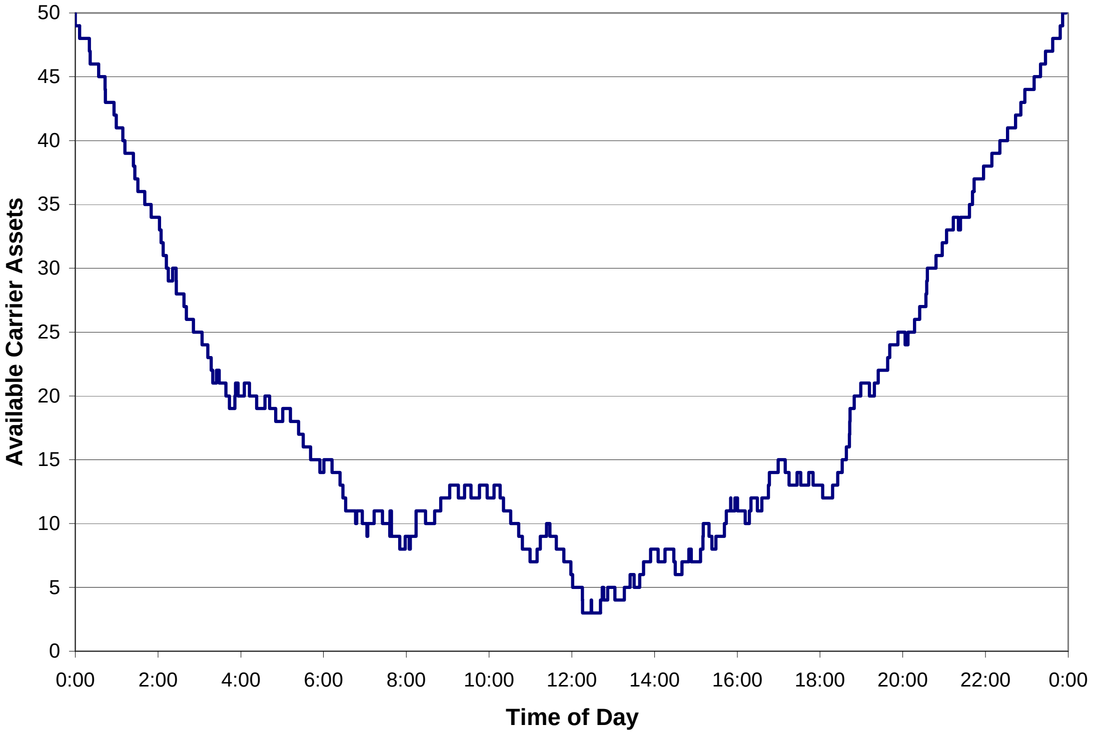


Figure 2–4: Available assets for a given carrier as a function of time of day

In order to model the impact of adding a military mission, we add a new departure when the asset leaves the hub and a new arrival when the asset returns to the hub. As long as the inventory never dips below zero, the asset can be "borrowed" without impacting the remainder of the schedule (assuming no delays or maintenance problems). However, if the inventory dips below zero, then that means a departure is scheduled at a time when no assets are available. Consequently, that departure will have to be delayed until an asset arrival.

As additional assets are volunteered, the marginal delay increases. Figure 2–5 shows the marginal delay from volunteering each asset for the entire day. The enterprise can then convert each delay into a cost that is proportional to the delay.

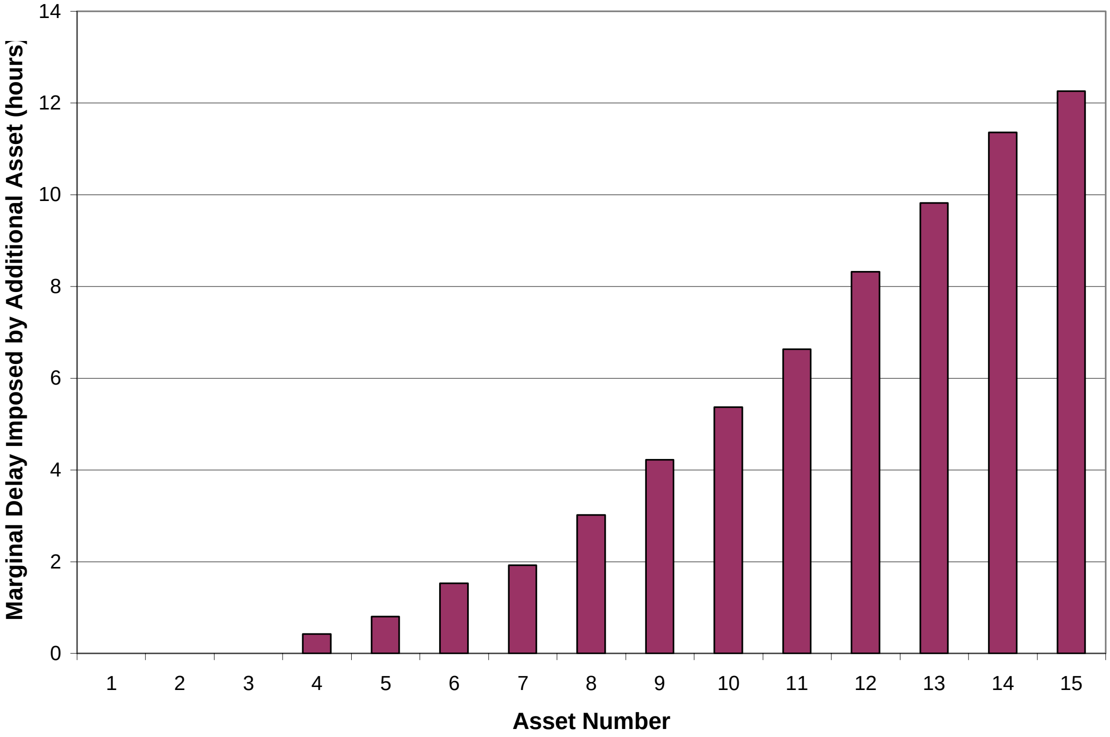


Figure 2–5: Marginal delay caused by volunteering each asset for a full day for a single enterprise from a single hub location

This micro-model of hub operations may seem unnecessarily complicated for computing a simple cost to provide an asset for a fixed amount of time. In particular, this operations model requires daily external demand data (to specify when assets depart and return) and detailed calculations to learn how much delay a particular asset adds.

To reduce this burden, we will use instead a simple, piecewise-linear cost function that describes the marginal cost of surrendering the $(d_{abt})^{\text{th}}$ available asset of type $a \in A_b$ that is owned by enterprise $b \in B$ for all of time period t (each time period could represent a six-hour block of time, for example). The piecewise-linear function is described by a series of doublets that specify a breakpoint and the projected slope from that breakpoint. For example, a function specified by the set of doublet pairs {(0,0), (3, 2000), (6, 500)} is equivalent to the following:

$$c_{abt}\left(d_{abt}\right) = \begin{cases} 0 & \text{if } d_{abt} \le 3 \\ 2000 \times \left(d_{abt} - 3\right) & \text{if } 3 < d_{abt} \le 6 \\ 6000 + 500 \times \left(d_{abt} - 6\right) & \text{if } 6 < d_{abt} \end{cases}.$$

This marginal cost function is zero, until the threshold $d_{abt} = 3$ is met, at which point the function becomes linear with slope 2000 for three units, and then linear with slope 500 for all $d_{abt} > 6$. If the asset is unavailable for several time periods, then take the sum of the costs over each period. As described in Appendix A.6, the initial set of piecewise-linear opportunity cost functions depend on the enterprise, the asset type, the location of the asset and the time period during the day (midnight-6am, 6am-noon, noon-6pm, 6pm-midnight). The same set of cost functions is used every day, but the argument passed to the piecewise-linear function (which includes how many assets have already been committed) may be different for each day and time period.

2.8. Infrastructure Capacity

In logistics, it is important to model the throughput capacity at the consolidation points. Some smaller foreign airfields, for example, may be overwhelmed by the arrival demand of a military airlift. In that case, it is critical that the flow of people and goods into the airfield be smoothed out as much as possible. Otherwise, predictable delays will occur and cause inefficiencies in scheduling and asset utilization.

We will not model this mathematically. However, it is an important consideration if the military is allowed to reject assignments proposed by a commercial enterprise. It can also be part of the central visibility into the operation given to each enterprise. In other words, United Airlines and American Airlines may both want to satisfy the same movement requirement. One useful tie-breaking rule would choose the airline that can deliver the payload on the least congested day.

2.9. Summary of VTC REF Results

As mentioned previously, four TASK research groups worked on the VTC REF through January 2002, when the program focus changed. There were several new results that were developed during FY01, which we summarize below.

- *Cornell University / University of Washington*: Developed the first formal measure of fairness for resource allocation (Lexicographical Min-Max fairness); Developed a detailed worst-case complexity analysis of the VTC domain based on various structural properties.

- *Stanford University*: Proved the optimality of a technique for fair imposition of tasks with private information; Developed a bidding clubs technique for collaboration among participants in a task allocation setting.

- *University of Texas at Austin*: Implemented the Sensible Agent testbed, which incorporates a fault-tolerant decision-making framework, into the VTC domain; Developed the first proven and validated algorithm for dynamic reorganization of decision-makers based on situational context.

- *Metron, Inc.*: Developed a collaborative auction plus mission swapping framework in which opportunity costs and controllable operating costs were cut by 50 percent over a centralized approach for single mission auctions.

For FY02, the research teams had proposed new ideas for investigation, and the REF would have expanded to add the University of Michigan team. The FY02 goals are listed below for completeness, even though the research was not performed.

- *Cornell University / University of Washington*: Characterize the structure of the effective complexity of combinatorial auctions; Apply randomization techniques to provide a super-linear speedup on hard combinatorial optimization instances.

- *Stanford University*: Exploit the effective complexity of combinatorial auctions (from Cornell/UWash) to design provably optimal mechanisms for fair task allocation where tasks may be complementary or substituted.

- *University of Michigan*: Apply the Cornell randomization techniques to improve the Disjunctive Temporal Problem algorithm efficiency by a factor of ten; Develop the first computationally feasible metric for agent update cost under complex temporal constraints.

- *University of Texas at Austin*: Analyze the scaling behavior and solution robustness of the Sensible Agent testbed as the environment and data uncertainty changes;

Incorporate the University of Michigan's agent update cost metric into the Sensible Agent testbed.

- *Metron, Inc.*: Leverage combinatorial auction strategies to cut the carrier opportunity cost and controllable operating cost by another 50 percent over the FY01 results; Identify other military transition opportunities for the VTC REF technologies.

In the next chapter, we describe the collaborative airlift planning approach that we developed in greater detail and present a series of experimental results that are summarized above.

3. COLLABORATIVE AIRLIFT PLANNING

In this chapter, we apply negotiation protocols to the distributed optimization problem of commercial air carriers supporting military airlifts, a problem area that: (1) is extremely relevant, timely, and has huge potential military payoff, and (2) has a rich environment for testing multi-agent systems. The operational goal is to make next-generation airlift procurement agreements more flexible and mutually beneficial without relying on centralized mechanisms that ensure convergence but reduce efficiency.

In Section 3.1, we describe the protocols and distributed optimization applied to the collaborative airlift planning problem and in Section 3.2, we present experimental results for a sample airlift scenario.

3.1. MULTI-AGENT VTC COLLABORATION PROTOCOLS

We propose an airlift procurement approach that uses software agents representing the commercial and military parties to collaboratively plan the airlift. Rather than have the military assign the missions arbitrarily according to obligation, the missions are auctioned to the highest bidder subject to a reserve price. Carriers can also exchange missions with one another when there is mutual benefit.

This collaborative approach provides more flexible planning, more reliable missions, and uses commercial "best practices" without integrating those practices into military planning systems or having to share those practices with the other carriers. This multi-agent approach satisfies the following properties:

- Allows a commercial carrier to assert its private, competitive interests through negotiation agents rather than having to make those interests explicit and public (as would be the case for classical, centralized optimization approaches);

- Provides incentives to carriers to volunteer assets early in the planning process;

- Enables the military agents to enforce airlift constraints such as delivery time windows and airfield congestion when evaluating offers from carrier agents;

- Enforces fairness in that no carrier can be forced to provide more than its airlift obligation, but carriers who want additional business may request it; and

- Guarantees that each mission is assigned to one of the carriers.

The remainder of this section describes the process by which a carrier optimizes its missions, the auction framework for allocating missions, and the process by which carriers swap missions.

3.1.1. Mission Planning by Individual Carriers

In order for carriers to optimize the missions for which they are responsible, we need to describe the characteristics of the airlift missions and the economic models used to represent carriers. We will only use a subset of the data sets described in Chapter 2 and Appendix A.

Airlift Demand. We represent the military contingency demand with a stripped-down TPFDD. This modified TPFDD is a set M of individual movement requirements. Associated with movement $m \in M$, we define

PAY_m = the payload vector (passengers, bulk cargo) for the movement;

POE_m = the origin or Point of Embarkation (POE) of the movement;

POD_m = the destination or Point of Debarkation (POD) of the movement;

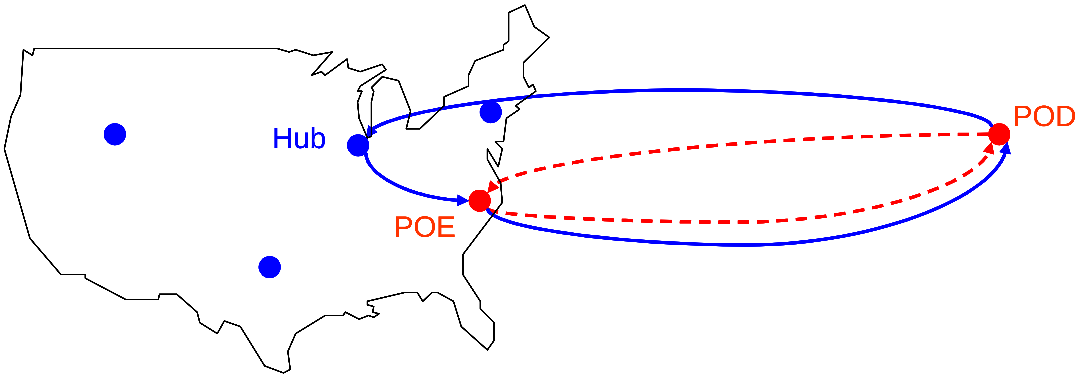
TT_m = the expected travel time between the POE and POD of the movement;

EAD_m = the earliest arrival date (EAD) of the movement at the POD; and

LAD_m = the latest arrival date (LAD) of the movement at the POD.

The payload consists of either the number of personnel or amount of bulk cargo (expressed in short tons). Due to commercial aircraft configuration (such as door size and floor strength), we assume that commercial carriers cannot move oversized and outsized cargo. We also assume that each movement will be ready to load at the POE at a date consistent with the EAD and LAD (typically, the EAD and LAD are about three days apart). To simplify matters, we assume that movement requirements have been aggregated or broken into either passenger or cargo missions, roughly the size of a wide-body commercial aircraft for each payload type when possible.

Carrier Economic Model. We assume that the cost for a carrier to assign an aircraft to satisfy a mission depends on two factors: (1) the location of the aircraft when it is pulled from service, and (2) the time interval during which the aircraft is unavailable for normal commercial flights. For simplicity, we assume that each carrier has a set of hubs with available aircraft and that aircraft must return to its original hub after completing a mission.



- **Revenue** (broken line) is paid by the military per passenger-mile or short ton-mile based on the round-trip distance between the POE and POD
- **Operating Cost** is proportional to the total distance traveled by the aircraft (Hub to POE to POD back to Hub)
- **Opportunity Cost** is the potential profit lost by satisfying military missions rather than commercial customers (cost of delays, etc.)
- **Profit = Revenue - Operating Cost - Opportunity Cost**

Figure 3–1: Components of mission profit for air carrier

Figure 3–1 shows the two components to the mission cost for the carrier: (1) the operating cost, which is the cost to operate the aircraft for that mission (carrier hub to POE to POD and back to hub), and (2) the opportunity cost, which is the potential profit lost by not having the aircraft available for commercial demand. The details of computing the opportunity cost are discussed in Section 2.7.

When a carrier is assigned a mission, it attempts to minimize the sum of the mission operating and opportunity costs through two choices: (1) which hub to use, and (2) what time the aircraft will depart the hub and return, subject to the mission EAD and LAD constraints.

3.1.2. Auction Framework for Mission Allocation

Figure 3–2 illustrates the different entities and interactions between entities in the airlift auction, which is a multi-threaded, Java simulation. Each buyer agent (representing one of the commercial carriers) and the seller agent (representing the military) acts on its own processing thread. Frequently, we refer to the buyer and seller agents as carrier agents and military agents, respectively. The agents use a messaging protocol developed by Metron to communicate with each other and with the auction and assignment modules.

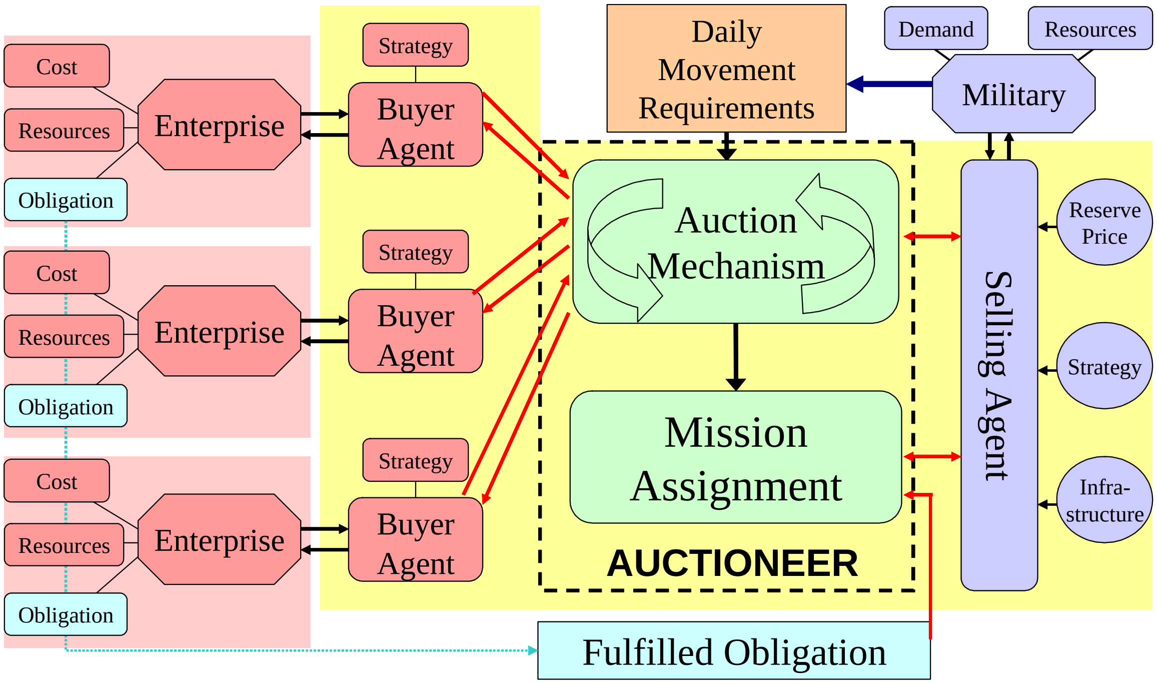


Figure 3–2: Description of airlift auction simulation architecture

Given a set of airlift missions to be performed, the auction module puts each mission up for bid sequentially by sending a message to each carrier agent specifying the details of the mission. To reduce profiteering, the military agent sends to the auction module a reserve price for the mission (that is not made public) to limit how much the military will pay.

In the experiments, the reserve price for a mission is set to be a multiple of the price that the military would pay if the mission was assigned (a fixed per-mile rate times the round-trip distance from POE to POD). If the reserve price is equal to the assignment price, then the military is guaranteed not to pay more for the airlift under the auction approach than it would under the assignment approach.

Each carrier agent represents a single carrier and has access to that carrier's fleet information, operating and opportunity cost functions, and CRAF obligation that specifies what fraction of the airlift the carrier is obligated to provide. As an incentive to volunteer assets early in the planning process, once a carrier has fulfilled its fraction of the airlift voluntarily, it has no residual military obligation. Under this protocol, a carrier benefits from negotiating its airlift assignments early in the planning process. The alternative is waiting until the attractive movements have been satisfied by other carriers and having to fulfill its obligation with the remaining missions.

For a given mission, each carrier agent computes an appropriate bid based on its private bidding strategy module. A bid consists of a price to perform the mission and the time at which the mission will be completed. In the experiments, the bid is based solely on the carrier cost to satisfy the mission. However, a carrier could make its bidding strategy more sophisticated by taking into account CRAF obligation or estimates of other agents' bids.

The auction mechanism module uses a Vickrey auction format with simultaneous sealed bids and a reserve price specified by the military agent. Under the Vickrey format, the lowest bidder wins, but receives the second-lowest bid amount. Although the military pays more than the lowest bid, the military benefits because the Vickrey format has desirable theoretical properties that encourage accurate bids [Vic61]. The winning carrier receives more than its bid, and each bidder benefits by not wasting effort trying to outguess its opponents.

The following condition ensures fairness and encourages realistic bidding: if all bids exceed the reserve price, then assign the mission to the carrier who has satisfied the smallest percentage of its CRAF obligation. In other words, the reserve price caps what the military is willing to pay for a mission. If no carrier is willing to accept that price, then the mission goes to the carrier who has satisfied proportionately the least. This encourages carriers to bid aggressively, even taking small losses on some missions to protect against large losses on missions that no carrier wants.

Ultimately, the military has final control over the airlift assignments. For example, the military agent may reject a bid due to excess congestion at the POD airfield on the day that the bidder proposed. This increases the cost to the military, but the military has that discretion.

3.1.3. Mission Swapping among Carriers

Carriers will occasionally "regret" winning a mission when a more suitable mission appears later in the auction. To remedy this, we allow carrier agents to swap missions with one another, as long as both carriers benefit from the swap. Figure 3–3 describes an algorithm for performing one-to-one (pairwise) swaps between carriers. A similar approach is used for one-way swaps in which both carriers benefit by one carrier giving the other a mission without receiving one in return.

1. All carriers generate a list of potential swaps of missions it owns with missions it does not own, sorted by the profitability of each swap.
2. The system selects a carrier to propose a set of swap requests using a random permutation.
3. The selected carrier proposes its most profitable swap to the carrier who owns the other mission. The other carrier responds in one of two ways depending on swap profitability:
   a. The other carrier rejects the swap if it does not satisfy the carrier's minimum profitability threshold. The first carrier then returns to step 3 using the next-highest profitable swap. If no other profitable swaps exist, then return to step 2.
   b. The other carrier accepts the swap, and both carriers exchange missions and update schedules. Since the selected carrier has had a swap accepted, return to step 2 to select the next carrier to propose a swap.

- The one-for-one swapping procedure ends when all carriers have proposed all profitable swaps and no swaps are accepted.

Figure 3–3: One-for-One mission swapping algorithm from one carrier's perspective

3.2. EXPERIMENTS

Data Set Characteristics. Table 3–1 lists several key characteristics of the data sets used in the following experiments, broken down into bulk cargo and passenger components. The modified TPFDD is based on a Desert Shield/Desert Storm-type scenario generated by USTRANSCOM for the DARPA Advanced Logistics Program. Additional guidance regarding rough parameters for airlift planning (such as turnaround times for refueling, unloading, etc) was provided by the Air Force *Air Mobility Planning Factors* pamphlet [USAF98].

The base revenue rate specifies how much the military would pay for a mission under an assignment solution. In these experiments, the auction reserve price is set to a multiple of the base revenue rate. In practice, the military likely would choose the reserve price during the planning of the airlift. In Section 3.2.2, we show that large reserve prices have little effect on the total revenue paid by the military when carriers are bidding truthfully.

	CARGO MISSIONS	PASSENGER MISSIONS
Total Commercial Payload	182,944	436,676
Number of Days for Airlift	90	90
Number of Carriers	14	11
Total Fleet Size	1143	1852
Number of Carrier Hubs	24	21
Aircraft Payload Capacity	70	262
Number of Missions	2753	1857
Base Revenue Rate	$0.2725/short-ton-mile	$0.0672/pax-mile
Operating Cost Rate	$15.29/mile	$13.07/mile
Opportunity Cost of Delay	$75/minute of delay	$75/minute of delay

Table 3–1: Parameters for collaborative airlift experiments

To simplify the planning process, all carriers use the same type of generic wide-body aircraft for each payload type, and TPFDD movement requirements were pre-aggregated into missions compatible with the wide-body payload capacity when possible. The carriers, fleet sizes and CRAF obligations are based on those of the 1999 CRAF carriers. We constructed the set of hubs used by each carrier, as well as the opportunity cost functions, by hand.

If this concept of operations was adopted to plan real airlifts, then the fleet characteristics and economic factors would be specific to each carrier and kept private from the other entities, including the military. The list of airlift missions and duration of the airlift would be maintained by the military and shared with the carriers only at the time of the auction.

The software agents would be distributed across a private communications network across which messages regarding mission information, carrier bids and auction results would be exchanged. The goal is to preserve the greatest amount of autonomy for each agent, while ensuring that the mission allocation process is fair to all players and guarantees a feasible, reasonably priced solution for the military.

3.2.1. Cost Comparison of Auction versus Assignment

In the first set of experiments, we compare the assignment results with those of the reserve price auction and the auction plus swapping. For the assignment run, each mission is assigned in sequence to a single carrier accordingly to CRAF obligation. This assignment is equivalent to a reserve price auction with a reserve price of zero. The carrier then chooses the hub and departure time that minimizes the cost to satisfy the mission.

The reserve price auction run uses a reserve price equal to 0.9 times the base revenue rate (other multiples are considered in Section 3.2.2). Missions that do not satisfy the reserve price are assigned to the carrier who has satisfied the smallest percentage of its CRAF obligation. After the auction ends, carriers may perform one-way or two-way mission swaps when mutually beneficial.

Even an optimal solution (in which every aircraft starts at and returns to the POE, rather than the carrier hub) has a large operating cost, so we want to measure the amount of excess operating cost above this optimal bound because that is what the carrier can control. We define the *Controllable Operating Cost* to be the difference between the operating cost for a given run and the optimal operating cost.

Figure 3–4 breaks down the results of the three runs (Assignment, Auction, and Auction plus Swapping) by passenger and bulk cargo missions. In each case, the auction reduces the operating and opportunity costs by at least 30 percent. Allowing carriers to swap missions drops the costs by another 30 percent, leading to a total cost reduction of over 50 percent compared with the Assignment solution for both cost categories and both payload types.

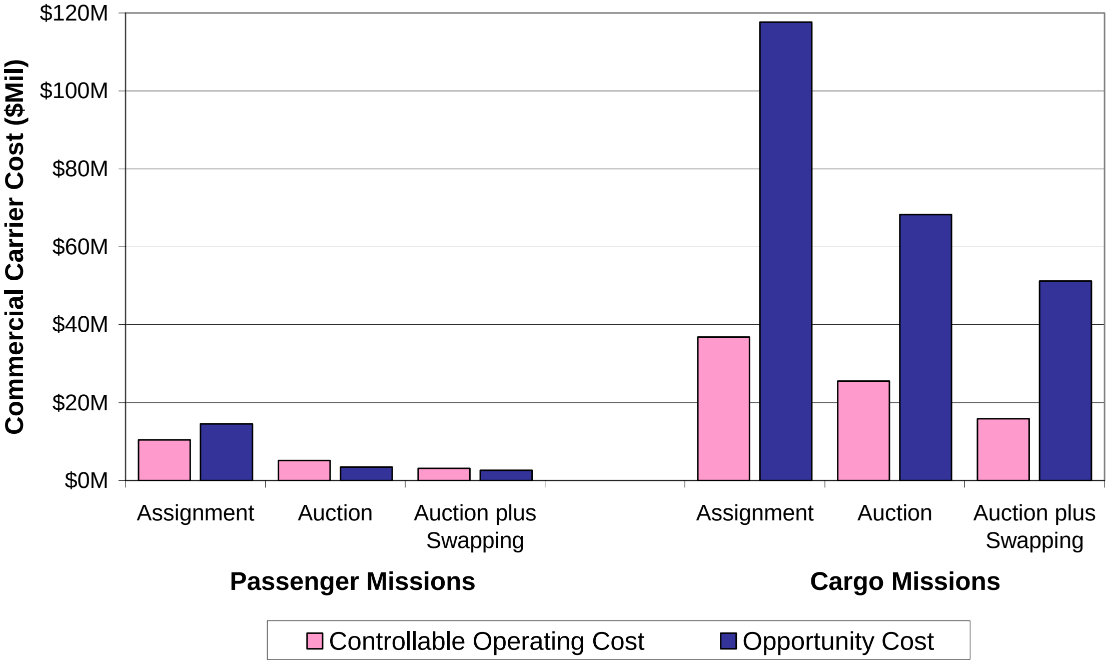


Figure 3–4: Carrier cost breakdown by protocol

3.2.2. Effects of Reserve Prices on Cost and Revenue

The sole focus of the remaining experiments is on cargo missions because that is the harder problem of the two in this airlift scenario. This experiment measures the effect of the reserve prices on the airlift solution using two pairs of runs. The first pair contains the assignment and the assignment plus swapping results for a series of five multipliers to the base revenue rate (0.6, 0.7, 0.8, 0.9 and 1.0). The second pair contains the reserve auction and the reserve auction plus swapping results for seven different reserve price multipliers to the base revenue rate (0.6, 0.7, 0.8, 0.9, 1.0, 1.2 and 1.5).

Figure 3–5 illustrates the four data series corresponding to the two pairs of runs. Starting with the assignment results, the revenue multiplier does not affect the cost of the assignment-only solution because the carriers try to minimize the cost of the same assigned missions. Swapping the assigned missions decreases the carriers’ airlift cost, but does not affect the revenue (which is fixed). Notice that the swaps become more attractive and effective as the revenue multiplier increases because the potential profit for carriers who have available capacity also increases.

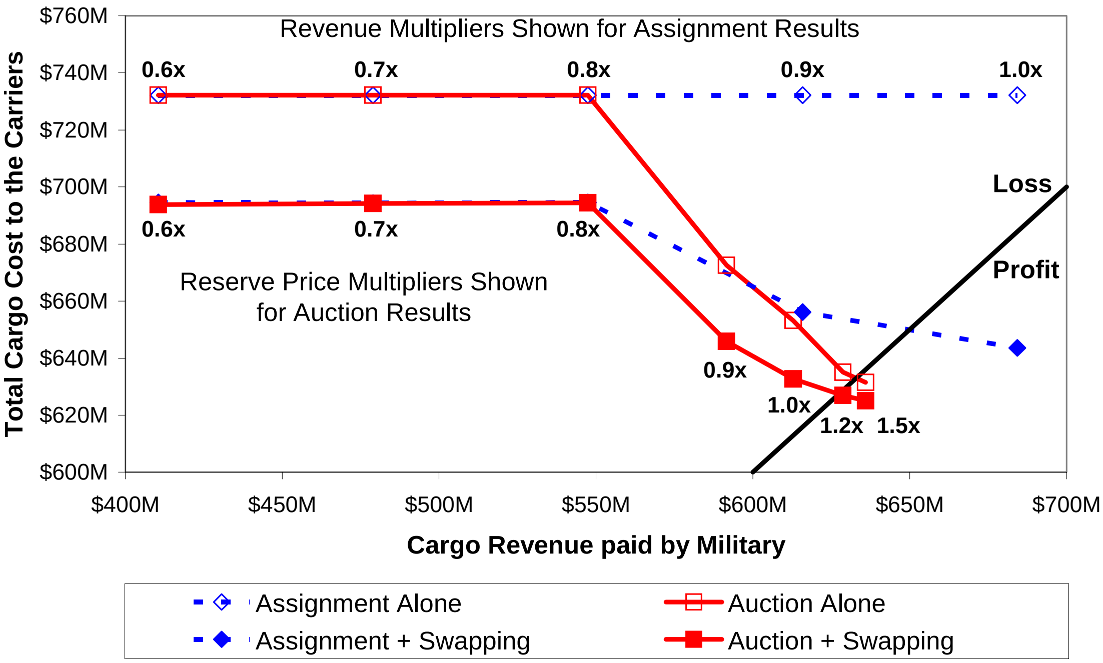


Figure 3–5: Carrier cost versus revenue paid by military for different reserve prices

The auction solutions show a different pattern. Up to a reserve multiplier of 0.8, almost no bids satisfy the reserve price, so the auction solution is essentially the same as the assigned solution. As the reserve multiplier increases, so does the number of carriers who can satisfy the missions efficiently. Consequently, the airlift cost drops. However, the rate of cost improvement slows as the multiplier increases because once one carrier bids less than the reserve price, increasing the reserve price will not lower the cost any further. Similarly, the Vickrey format means that the revenue increases slowly as the reserve multiplier increases because the military pays the lesser of the reserve price and the second-lowest bid.

The Vickrey mechanism also caps a carrier's profit for a particular mission. If all carriers have roughly the same cost structure with the primary differences being hub locations and opportunity costs, then bids will tend to cluster over a relatively small range. Consequently, the small gap between the lowest bid (lowest cost to perform the mission) and the second-lowest bid (the revenue that the military will pay) means a small profit for the winning carrier. However, when bids do not satisfy the reserve price, then there can be a large gap between the military's reserve price and the cost to the carrier who is assigned the mission.

Unlike the assignment results, post-auction swapping becomes less effective as the reserve multiplier increases. The reason is that a carrier who wins an auction with the lowest bid tends to be best-suited to perform the mission, so there is no incentive to swap.

One obvious question is what reserve price should the military choose in a given instance? The answer is that the reserve price should be set low enough to induce active, accurate bidding from the carriers, but no lower. If the reserve price is set too low, then a higher proportion of missions will be assigned. This leads to unprofitable missions and discourages carriers from CRAF participation in the future. On the other hand, if carriers are bidding reasonably, then the reserve price can be set high because the Vickrey format limits the price that the military pays.

Potential for abuse appears when carriers exploit a reserve price that is set too high by agreeing collectively to bid artificially high. If all carriers agree to inflate their bids by 20 percent above actual cost, then the same carriers will win the auctions, but the revenue will be 20 percent higher. However, there is an incentive for one carrier to violate this collusion by bidding accurately. If that carrier wins the auction, then the military pays the carrier based on the second-lowest bid (which is inflated). As with Prisoner's Dilemma problems, this bidding exploit fails if more than one carrier violates this agreement. To discourage "gaming" the system, the military may choose not to share the reserve price with the carriers.

3.2.3. Computational Effort and Unfair Swapping

Figure 3–5 raises two other interesting points. The first is that swapping the assigned missions produces a relatively low-cost solution without the overhead and infrastructure of an auction. The second is that the results assume swapping when there is mutual benefit. What would be the impact of considering an unfair swapping approach in which swaps are accepted as long as there is a net benefit (one carrier could be slightly worse off, but the other carrier much better off)? Although impractical, this result would provide a useful bound on the fair swapping solution.

To address the first point, swapping does alleviate much of the burden associated with the assignment, but there are two primary disadvantages. The first is that the amount of improvement is sensitive to the revenue rate that the military chooses. Ideally, the military would like to pay only enough to allow the carries to break-even or make a small profit. However, under the assignment, the revenue paid increases linearly with the revenue rate, while the cost decreases due to swapping. Conversely, the auction results are not as sensitive to the reserve price, especially for high values, because the Vickrey mechanism acts as a natural cap on profits.

The other disadvantage of swapping assigned missions is the greater computational effort (number of swaps) required to reach the final solution, compared with the auction approach. There are three reasons why this effort is required: (1) there is a large list of potential swaps that would be attractive to a particular carrier; (2) most proposed swaps are rejected as unprofitable by the other carrier; and (3) after a number of swaps are accepted and executed, the profitability of the remaining list of potential swaps must be re-evaluated.

The left-hand side of Figure 3–6 (up to where the data series are marked “Start Unfair Swapping”) shows the consequences of this computational effort. Using a reserve multiplier of 0.88 (and a revenue multiplier such that the assigned revenue would be the same as the auction revenue), we plot the total airlift cost as a function of cumulative CPU time.

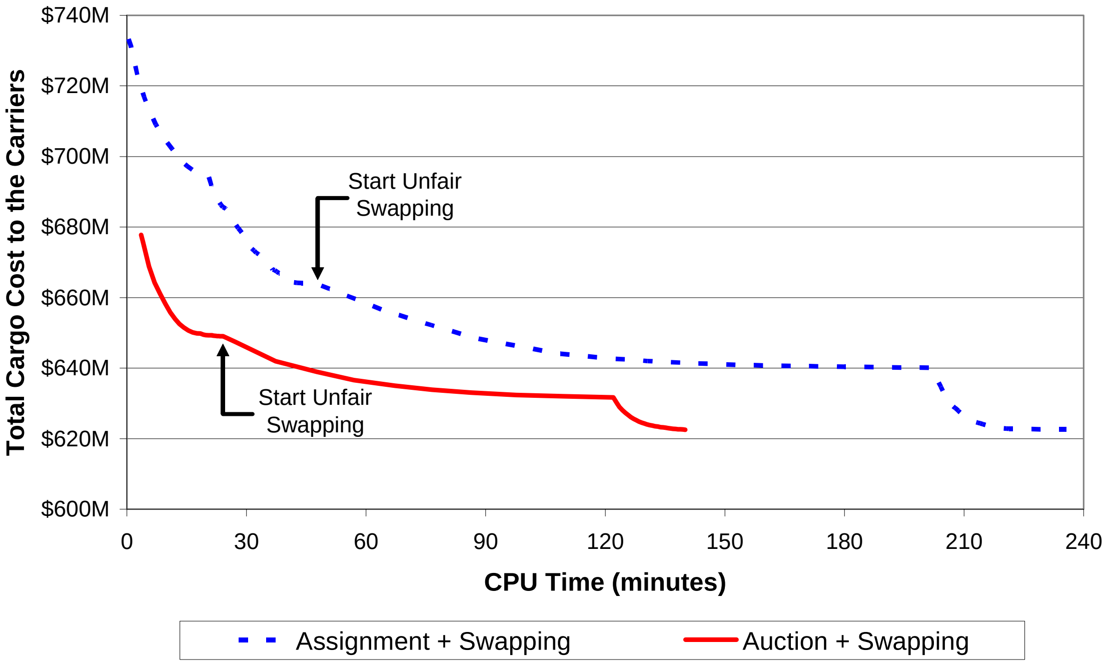


Figure 3–6: Effect of unfair swapping on auction and assignment solutions

Although the initial assignment is nearly instantaneous, another 30 minutes of swapping is required to lower the cost to be equal to what the auction reaches in less than four CPU minutes. In fact, the assigned swap solution converges after about 45 minutes at a cost comparable to the auction plus swap after seven minutes. Note that the computational burden is distributed across fourteen carriers, so a total CPU time of 15 minutes corresponds to about one minute of real time.

To address the latter issue of the unfair swapping bound, we start with the fair swapping solutions and perform unfair swapping in which there only needs to be a net benefit to the system in order to accept a swap. The reason that this is unfair is that each mission tends to end up with the carrier who can satisfy the mission at minimal cost, regardless of profit or obligation.

Figure 3–6 shows that both solutions converge to roughly the same cost through unfair swapping. The drop at 120 minutes for the auction solution and 200 minutes for the assigned solution is due to a switch between considering two-way swaps and one-way swaps. Although roughly the same solution is reached in the limit, the assigned solution requires nearly double the CPU time (and double the number of swaps) to converge to the auction solution.

3.3. Collaborative Airlift Planning Conclusions

We presented a distributed optimization approach that uses software agents – representing the interests of the military and commercial carriers – to collaboratively plan an airlift using commercial aircraft. By auctioning the missions subject to a reserve price and allowing carriers to swap missions when mutually beneficial, this approach cut the controllable operating costs and schedule disruption costs by more than half compared with a centralized assignment approach. Furthermore, this new approach to airlift procurement protects the military by capping mission profit potential using a Vickrey auction mechanism, and protects the carriers from being forced to share information or cooperate with its economic competitors. In the future, we would like to investigate more sophisticated bidding strategies for the carriers and expand the auction to allow concurrent auctions and bids on bundles of more than one mission.

In the next chapter, we present the UAV coordination problem that became the focus of the TASK research after 11 September 2001.

4. COORDINATED UAV SURVEILLANCE (TARGET MONITORING)

As the size of unmanned aerial vehicle (UAV) fleets increases in the future, so will the need to coordinate these fleets effectively. In this chapter, we define several negotiation mechanisms for autonomous, distributed coordination of surveillance tasks in which the goal is to maintain position estimates on a number of known targets. The problem of detecting a set of targets with unknown locations (all that is known is a probability distribution for each target location and an estimated motion model for each target) is considered in Chapter 5 (target search).

These surveillance mechanisms are based on dynamic target swapping between UAVs, in which the criterion for swapping can be greedy or cooperative and where the amount of information shared by UAVs can be relatively high or low. The results show that high-quality system solutions can be obtained through local optimization by individual UAVs. In addition, we show how the rate of convergence to good system solutions can improve given cooperative UAV behavior (adherence to system goals rather than strictly local goals) and greater information sharing.

4.1. INTRODUCTION

Consider the dynamic problem of a fleet of *M* UAVs performing surveillance on *N* mobile targets over a fixed area. When a UAV passes over a target, the UAV sensor updates the target position estimate. The fleet objective is to maintain tight location estimates on the set of moving targets over time, and the UAVs satisfy this objective by visiting each target as frequently as possible.

One solution approach is to assign a subset of targets to each UAV and then each UAV solves a traveling-salesperson problem (TSP) to construct a tour that minimizes the revisit time for targets in that tour. However, partitioning the target set into subsets that lead to balanced, compact tours for each UAV is difficult. Furthermore, the optimal partition changes over time as targets move, targets are added or removed from the tasking list, or UAVs enter or leave the system.

The focus of this chapter is on developing dynamic negotiation mechanisms (i.e., swapping strategies) for allocating targets to UAVs, not on optimizing the TSP tours. Instead, we rely on standard tour construction and improvement heuristics such as those described in Johnson and McGeoch [JM97].

Given an initial set of assigned targets, each UAV uses a sweep heuristic to construct an initial tour. Start by converting the last known target location to polar coordinates, using the average target location as the center of the coordinate system. To construct the heuristic initial tour, the UAV picks the closest target to start the tour and then visits the remaining targets in increasing order of their angular coordinate. For example, if the closest target has angle 56°, then the targets are visited in increasing order from 56°; that is, {56°, 75°, 128°, 235°, 320°, 20°}.

To improve the tour (identify shorter tours), the UAVs use a 2-opt heuristic, which is an iterative way to break and reconnect tours to search for improvements. The 2-opt heuristic breaks two tour edges and reconnects the tour by flipping the broken sequence. For example, if the original order was {1-2-3-4-5-1}, then 2-opt tours include {1-3-2-4-5-1}, {1-2-4-3-5-1} and {1-4-3-2-5-1}. If any tour is shorter than the original, then make the change permanent. Otherwise, continue 2-opt combinations for a fixed number of iterations or until no improvements are found.

Figure 4–1 illustrates the impact of dynamic negotiation using one of the cooperative swapping strategies from section 4.3. The "Before" picture shows a set of UAVs with an initial target assignment. Each UAV solves a TSP to minimize its tour. However, the targets are spread out over space causing long tours and relatively infrequent target revisits. The "After" picture shows the same set of targets after several target swaps have been performed by UAVs, leading to much shorter tours.

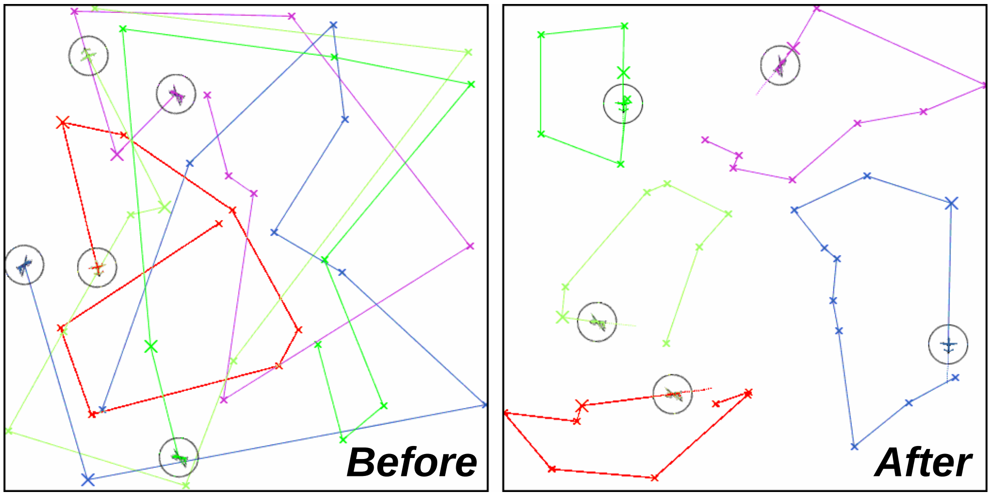


Figure 4–1: Illustration of UAV tours before and after performing cooperative target swapping

In the following sections, we describe a set of negotiation mechanisms by which UAVs swap targets with each other in order to improve the solution either locally (shorter tours) or globally (reduce the squared target location errors). Section 4.2 describes a greedy approach in which UAVs propose one-for-one target swaps that can only be accepted if both UAV tours decrease. Section 4.3 generalizes this swapping mechanism to include uneven (e.g., one-for-none) swaps and relies on cooperative behavior (focus on system goals rather than local goals) from the UAVs. Section 4.3.4 presents experimental results that compare the different swap strategies.

4.2. Greedy Target Swapping

In all of the surveillance negotiation mechanisms that we describe, we assume that UAVs share information with each other regarding target location updates from the UAV sensors. UAVs are also aware of which UAV owns or has responsibility for each target. Some mechanisms will require additional information to be shared, and we will make these requirements clear in the descriptions.

The first mechanism (or swapping strategy) that we describe is called “Greedy Even”. The proposal type is an even, one-for-one exchange. That is, one UAV proposes to exchange a particular one of its targets for a particular target from another UAV. The greedy part is that the proposed swap must shorten the tours of both of the UAVs to be accepted.

4.2.1. Greedy Even strategy

At each time period, one UAV will be selected to make a proposal, call it UAV1. Divide the set of all targets into two sets, J_1 and J_*, representing those targets owned by UAV1 and not owned by UAV1, respectively. Consider the set of all proposal combinations $(j,k) \in J_1 \times J_*$. UAV1 can evaluate potential swap (j, k) by computing the new tour length associated with removing target j and inserting target k into the best of the $|J_1|–1$ tour insertion points. If the tour would be shorter given swap (j, k), then store this swap in the proposal list. Continue for all combinations, sorting the stored swaps by estimated decrease in tour length.

UAV1 then proposes the best one of these stored swaps to the owner of target k, call it UAV2. UAV2 then evaluates whether the (j, k) swap reduces its tour length by removing k and inserting j into the best tour insertion point. If the tour improves (becomes shorter), then the proposal is accepted, the targets are exchanged, and the other UAVs are notified of the change in target ownership.

In practice, we can improve the computational scaling of constructing proposals by considering only targets near the tour, rather than the full set $|J_*|$. Also, proposals at the top of the list that have recently been rejected may be bypassed for some time in favor of a lower-ranked proposal.

For a rough complexity analysis, there are order $o(N/M \times N)$ possible swap proposals, which can be reduced to $o(N/M \times N/M)$ proposals, in practice. Each proposal requires computing the tour length for $o(N/M)$ insertion points, and each tour requires $o(N/M)$ target to target distance calculations. Constructing a proposal, then, requires $o(N^4/M^4)$ distance calculations and evaluating a proposal requires $o(N^2/M^2)$ distance calculations, making the total complexity

$$\text{Greedy Even: } O\left(\frac{N^4}{M^4}+\frac{N^2}{M^2}\right). \tag{4.1}$$

The first term dominates because the number of targets, N, is generally much larger than the number of UAVs, M.

4.2.2. Greedy Experimental Results

We performed a set of experiments in a simulation area $A = 500\times500$ with $M = 10$ UAVs and $N = 50$, 100 or 150 targets. Each UAV has speed $u = 10$ units/time and a 10-unit sensor footprint, meaning that the sensor will detect any target within this footprint and update the target location after detection. The targets move according to a Pearson random walk model, which means that each target takes a step of length $v = 0.1$ in a uniformly random direction in each time period.

In each experiment, N targets are scattered uniformly and assigned to the M UAVs arbitrarily. For 3,000 time periods, each UAV traverses its tour without swapping. For the remaining 7,000 periods, UAVs take turns proposing swaps using the Greedy Even strategy, with one proposal per time period. The results for each value of N are averaged over ten independent trials.

Figure 4–2 shows the average location error per target over time for $N = 50$, 100 and 150 targets. This plot also includes a theoretical prediction of the average location error per target given “optimal” target assignments and tours. The prediction is based on asymptotic estimates of the optimal tour length of N points in a unit square and the diffusion properties of the Pearson random walk model. The derivation of the functional form for this prediction appears in Appendix C, and the asymptotic result for the average location error per target is reproduced from equation (C.8),

$$d_{Avg} \approx 0.58v\cdot\sqrt[4]{\frac{AN}{M^2u^2}}\,. \tag{4.2}$$

Figure 4–2 illustrates the convergence problems of the Greedy Even protocol as the number of targets increases while keeping the number of UAVs fixed. For a 5:1 target-to-UAV ratio, the tours converge within a few hundred periods to the predicted error level. However, as the ratio increases, the UAVs fail to partition the targets effectively because swap proposals are selected based only on the benefit to the proposer, but the evaluating UAV frequently rejects the swap as unattractive.

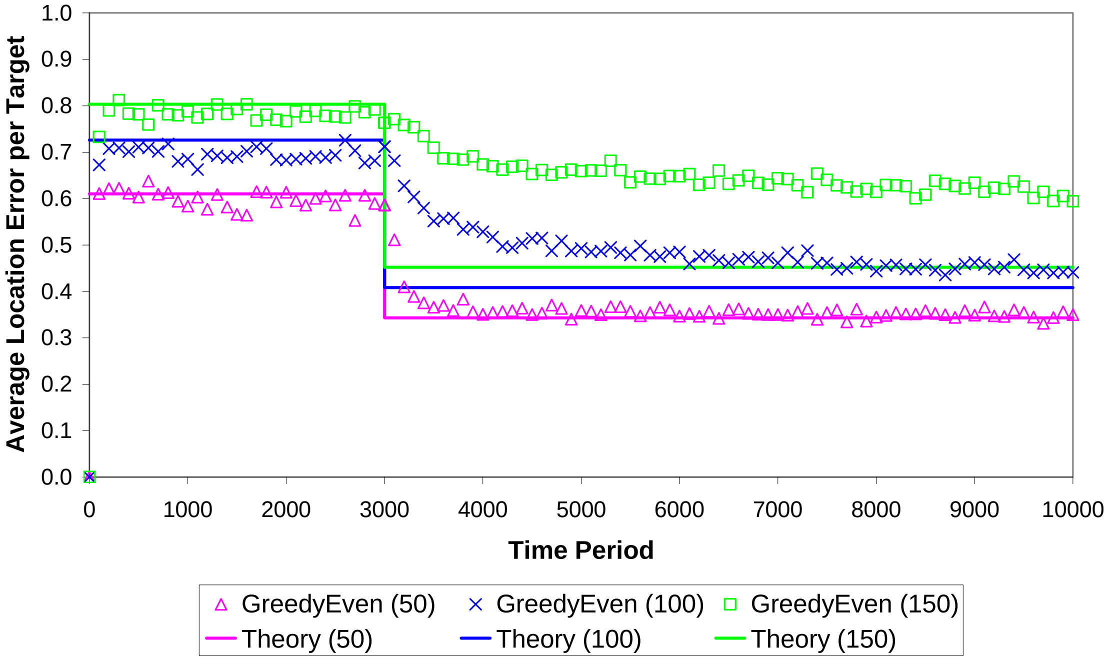


Figure 4–2: Average target error given no swapping for 3,000 periods, then Greedy Even for remaining time; performed using 10 UAVs and 50, 100 and 150 targets

4.3. Cooperative Target Swapping

The Greedy Even strategy is deemed "uncooperative" because proposing and evaluating swaps are based solely on satisfying local goals. For example, consider an even swap that decreases the first UAV's tour by 100 units but increases the second UAV's tour by two units. Under a greedy approach, the second UAV would reject the swap even though the system would have a net benefit.

In this section, we describe a new, cooperative decision rule for whether to accept a swap proposal. This cooperative rule takes into account the system goal of minimizing the sum of the squared target location errors. We perform a series of experiments that show the benefits of these "cooperative" strategies.

Consider a set of target assignments J_1 and J_2 and tour lengths of l_1 and l_2 for two UAVs. If a swap is proposed that would lead to assignments J_1' and J_2' and tour lengths l_1' and l_2', then the cooperative decision rule for evaluating that swap proposal is to accept only if

$$l_1' \cdot |J_1'| + l_2' \cdot |J_2'| \; < \; l_1 \cdot |J_1| + l_2 \cdot |J_2| \, . \tag{4.3}$$

Equation (4.3) is copied from equation (D.4) in Appendix D, which contains the full derivation of this decision rule. In the remainder of this chapter, we will refer to the quantity $l \cdot |J|$ as the “workload,” and the cooperative rule for judging swaps is to accept when the total workload decreases.

Equation (4.3) has several notable properties. First, if all UAVs have the same number of targets and only even swaps are proposed, then the decision rule simplifies to accepting the proposal when the sum of the tour lengths decreases. Second, uneven swaps (e.g., one-for-none) can be included, whereas in the greedy case, an uneven swap means that one of the tour lengths must increase and the swap is then rejected. Third, the number of targets owned by each UAV is a factor. Moving a target from a long tour that decreases slightly to a short tour that increases by a larger amount may be an acceptable swap depending on the number of targets in each tour.

Having stated the cooperative decision rule, we describe a series of three new swapping strategies that use this rule to evaluate swaps: Cooperative Even, Basic Push and Advanced Pull. Other swapping strategies can be derived from this cooperative approach, but the three that we considered are representative of the cooperative class of strategies.

4.3.1. Cooperative Even strategy

The first rule modifies the Greedy Even strategy to incorporate the total workload rather than making the swap evaluation decision based solely on the tour lengths for each UAV. The process for constructing swap proposals is similar, but the sorting of swap proposals needs a closer look.

Consider a sort based on the estimated change in the workload from equation (4.3),

$$l_1' \cdot |J_1'| + l_2' \cdot |J_2'| - l_1 \cdot |J_1| - l_2 \cdot |J_2| \, . \tag{4.4}$$

Since only even swaps are considered under this strategy, the target count for each UAV remains the same. That is, $J_1' = J_1$ and $J_2' = J_2$. Furthermore, the first UAV does not know what the new tour length would be for the second UAV after the swap, so we assume that UAV1’s estimate of UAV2’s tour length does not change. That is, we assume that UAV1’s estimate of $l_2' = l_2$.

By substituting these values of J_1', J_2' and l_2' into equation (4.4), we can compute the estimated change in total workload from UAV1's perspective,

$$l_1' \cdot |J_1| + l_2 \cdot |J_2| - l_1 \cdot |J_1| - l_2 \cdot |J_2| \;=\; (l_1' - l_1) \cdot |J_1| + (l_2 - l_2) \cdot |J_2| \;=\; (l_1' - l_1) \cdot |J_1|. \quad (4.5)$$

None of the even swaps proposals can change the number of targets owned by UAV1, so the sort is based on the change in UAV1's tour length as before. One difference from the Greedy Even protocol, though, is that swaps that increase UAV1's tour length may be proposed because the benefit to UAV2 may decrease the total workload.

Since the basic mechanics of Cooperative Even are similar to Greedy Even, so is the computational complexity

$$\text{Cooperative Even: } O\left(\frac{N^4}{M^4} + \frac{N^2}{M^2}\right). \quad (4.6)$$

The Cooperative Even strategy requires slightly more information sharing than Greedy Even. In order for UAV2 to judge whether to accept a proposal from UAV1, the proposal must include the targets to be swapping and the change in UAV1's workload if the proposal is accepted.

4.3.2. Basic Push strategy

The first uneven strategy that we consider is called a Push. A UAV proposes to "push" one of its targets to another UAV and receive nothing in return. The chosen target is the one that maximizes the reduction in the proposer's tour if removed. The proposal is sent to the UAV closest to the pushed target, along with the change in the proposer's workload if the swap is accepted.

This approach requires each UAV to know the locations of the other UAVs. If the target swap has been rejected recently by the other UAV, then the proposer can propose the target to another UAV or propose the next-best target for reducing its tour.

Choosing which target to propose requires $o(N^2/M^2)$ distance calculations, and computing the distance from each target to each UAV requires $o(N/M \times M)$ distance calculations. Evaluating a proposal requires computing the tour length for $o(N/M)$ insertion points, and each tour requires $o(N/M)$ target to target distance calculations. The total computational complexity, then, is

$$\text{Basic Push: } O\left(\frac{N^2}{M^2}+N+\frac{N^2}{M^2}\right). \qquad (4.7)$$

4.3.3. Advanced Pull strategy

The final cooperative strategy we consider is called Advanced Pull, in which one UAV proposes to "pull" a target away from another UAV and provide nothing in return. The "advanced" part involves additional information sharing between UAVs regarding the marginal decrease in workload associated with removing each of its targets.

This additional information allows the proposing UAV to have an estimate of the change in workload from the other UAV and to know in advance whether the deciding UAV will accept the proposal. When the proposal is sent, the proposer includes its change in tour length. If there are no targets that can be pulled to reduce the total workload, then the proposing UAV declines to propose.

The proposer could request marginal workload information for all $o(N)$ targets, but in practice, only a subset of $o(N/M)$ nearby targets are considered. After removing each of the $o(N/M)$ targets, compute the new tour length, which requires $o(N/M)$ distance calculations for each target. Selecting the best proposal by inserting optimally each of the $o(N/M)$ targets into the proposer's tour requires $o(N^2/M^2)$ distance calculations for each possible insertion. Evaluating a proposal requires computing a single tour length after removing the proposed target, which involves $o(N/M)$ distance calculations. The total computational complexity given these three computational components is

$$\text{Advanced Pull: } O\left(\frac{N^2}{M^2}+\frac{N^3}{M^3}+\frac{N}{M}\right). \qquad (4.8)$$

To compare the relative computational complexity of the three strategies, consider the case with M = 10 UAVs and N = 100 targets. For this case, the complexity of each strategy would be roughly o(10,000) for Greedy Even, o(10,000) for Cooperative Even, o(300) for Basic Push, and o(1,100) for Advanced Pull.

4.3.4. Cooperative Experimental Results

We performed three sets of experiments to compare the Greedy Even swapping strategies with the three cooperative swapping strategies. The first set of experiments uses the same parameter settings as in section 4.2.2 (Figure 4–2), except we replace the Greedy Even swapping strategy with Advanced Pull.

Figure 4–3 summarizes the experimental results. Not only do the Advanced Pull results converge more quickly once swapping begins than the Greedy Even results in Figure 4–2, it outperforms the theoretical prediction for 50 and 100 targets, and matches the prediction for 150 targets.

There are two primary reasons for this level of improvement. First, the Advanced Pull swap proposals are guaranteed to be accepted by the deciding UAV if the proposing UAV can identify a swap opportunity that would improve the system workload. Second, the UAV tours are not restricted to have the same number of targets over time. By allowing tours to have a variable number of targets, UAVs can take advantage of natural target clustering when and where it occurs.

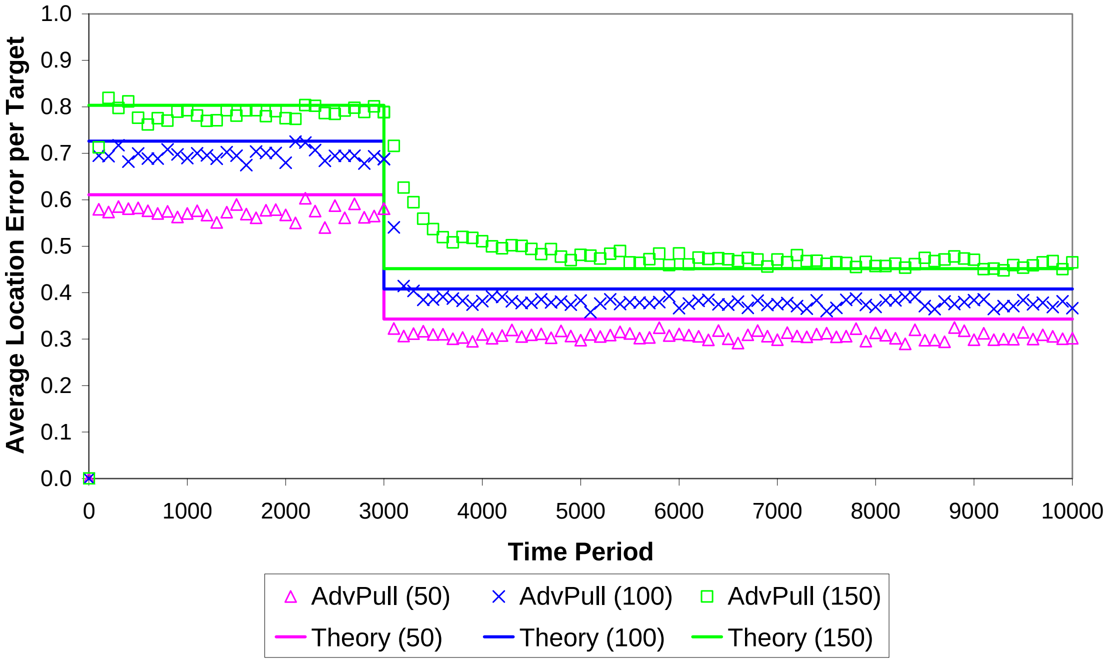


Figure 4–3: Average target error given no swapping for 3,000 periods, then Advanced Pull for remaining time; performed using 10 UAVs and 50, 100 and 150 targets

Since the cooperative strategies attempt to minimize the sum of squared target location errors, the remaining results will show the root-mean-squared (RMS) target locations errors rather than the average. In addition, we update the prediction models to estimate the RMS error rather than the average.

Figure 4–4 illustrates the RMS error for each swapping strategy given 10 UAVs, 150 targets, and swapping from the start of the simulation. The results are averaged across ten independent trials. Again, Greedy Even performs the worse, followed by Cooperative Even, Basic Push, and finally Advanced Pull, which nearly reached the predicted RMS error for an "optimal" partitioning of the targets.

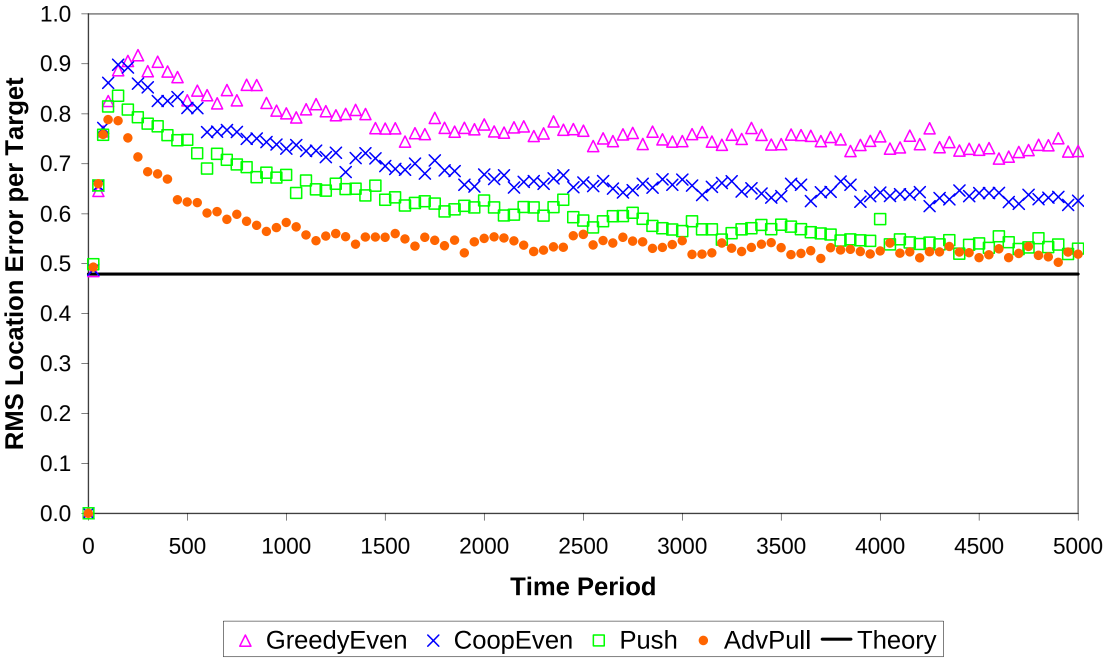


Figure 4–4: RMS target error for 10 UAVs and 150 targets using four swapping strategies; swapping starts immediately

For the final set of experiments, we ran each swapping strategy from the start of simulation against 10 UAVs and either 50, 100 or 150 targets. We averaged the RMS error over the final 5,000 periods of each 10,000-period run, and then averaged those results over ten independent trials. As Figure 4–5 shows, Advanced Pull and Basic Push track pretty well with the prediction model, but the gap increases with the number of targets. The Even strategies performed worse, and the errors diverge as the number of targets increases.

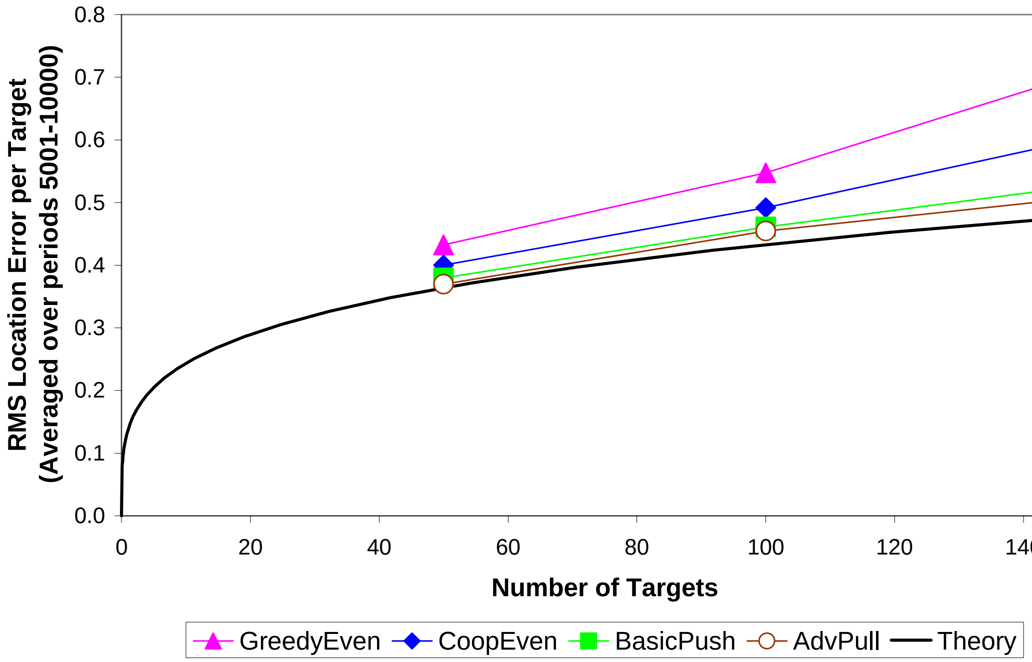


Figure 4–5: RMS target error for each swapping strategy averaged over last 5,000 periods of the 10,000-period runs

In conclusion, we derived in this chapter several negotiation mechanisms by which UAVs can swap targets with other UAVs in order to improve surveillance performance. The Greedy Even swap strategy required the least amount of information to be shared between UAVs, and allowed UAVs to act based on local goals rather than system goals. However, this greedy strategy scales poorly in performance and computationally as the number of targets increases.

The cooperative strategies that attempted to satisfy the system goal of minimizing the sum of the squared target location errors perform better in terms of dividing the targets across UAVs. The computation complexity varied with the particulars of the strategy, but all benefited from the cooperative rules for evaluating swap proposals and several were able to take advantage of additional information sharing requirements.

Furthermore, cooperative strategies that allow uneven swaps enable UAVs to exploit target clustering to further improve the system solution, to preserve solution quality as the number of targets increases, and to adapt quickly to changes in the number of UAVs and targets in the environment.

5. COORDINATED UAV SEARCH (TARGET DETECTION)

In the target surveillance problem, the UAVs negotiated target assignments with the goal of maintaining position estimation on a set of mobile targets. In the target search problem discussed in this chapter, UAVs coordinate their efforts searching for either stationary or mobile targets given probability distributions on the location of each target and an estimated motion model for each mobile target.

Under the coordinated UAV search protocols that we have developed, each UAV optimizes a local finite-horizon search path in real time and deconflicts the resulting search paths with the other UAVs. The optimization allows each UAV to perform an effective search, and the deconfliction ensures that the set of paths generated by the UAVs reduce redundant coverage. For example, two neighboring UAVs that optimize their paths independently can end up with search paths that are similar because both paths follow the same probability gradient (i.e., one UAV follows the other along a path that maximizes the probability of detection).

There are several advantages and disadvantages to the general class of local optimization mechanisms. The advantages include:

- Local optimization mechanisms use real-time target probability information and sensor reports effectively. In particular, local optimization approaches leverage both positive (target detection) and negative (no detection) sensor data to perform a thorough search for a nearby target.

- UAVs communicate pairwise with each other and require no third party broker. The UAVs typically share information about where each UAV has been, what sensor reports were collected at those locations, and what search path each UAV plans to take.

- Local optimization is well-suited to parallelization and is robust to disruption in communications because each UAV has sufficient local information to continue searching even without communicating (although not as effectively as when UAVs can communicate).

- Deconfliction can be generalized to have a single UAV coordinate the movements of multiple nearby UAVs. This coordinated approach would be more effective at containing an evasive target, for example.

However, local optimization approaches can also have undesirable properties, including:

- Choosing an appropriate planning horizon length for the search path (e.g., the UAV optimizes its search path over the next five moves) can be difficult. If the planning horizon is too short, then planning can became too myopic and miss opportunities that could be identified using a longer horizon. On the other hand, using a longer planning horizon may lead to better search decisions, but may be computationally expensive. For example, if a fixed-horizon plan consists of a sequence of UAV movements either left, straight or right at each time step, then the space of possible search plans grows exponentially in the number of time steps.

- Some forms of deconfliction may not converge or fail to converge to a desirable solution in a reasonable amount of time. For example, if two UAVs decide independently to visit the same cell in the next time step, which UAV should concede to reduce duplicative coverage? If both UAVs decide to concede, then the deconfliction could lead to a situation in which neither UAV visits a relatively high value cell in the next time step.

In the next section, we describe the Bayesian likelihood approach to target search used in this study. The emphasis is on how sensor information from the UAVs can be fused into the spatial probability distribution used to estimate the target position. In addition, we describe the random walk motion model used to model mobile targets. Subsequent sections describe the details of the finite-horizon search path planning approach and a series of experiments designed to evaluate the performance of the collaborative planning mechanisms.

5.1. Bayesian Likelihood Approach to Target Search

A Bayesian, nonlinear likelihood approach to target search is the foundation upon which we develop our collaborative UAV search planning protocols. Figure 5–1 illustrates the functional process flow that we will follow to perform an update under this Bayesian likelihood approach.

In this section, we focus on the first step and the final three steps in the functional flow. We describe the basis for the target probability maps, apply a motion model for the moving target models, and then fuse sensor information into the target probability map using likelihood functions. The second and third steps, optimizing the search path and deconfliction, will be covered in section 5.2.

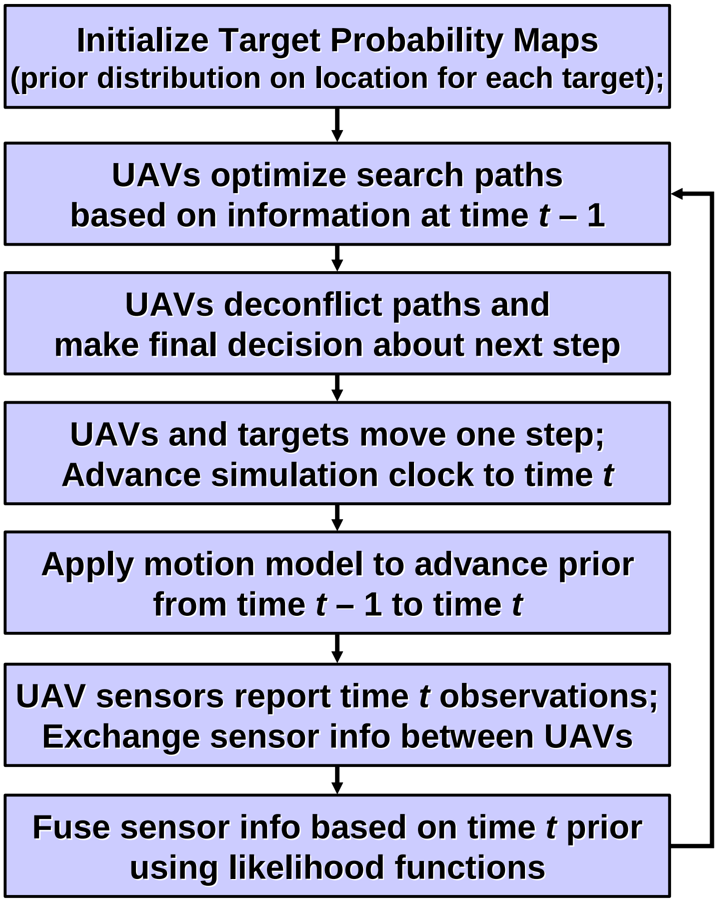

Initialize Target Probability Maps
(prior distribution on location for each target);
UAVs optimize search paths
based on information at time *t* – 1
UAVs deconflict paths and
make final decision about next step
UAVs and targets move one step;
Advance simulation clock to time *t*
Apply motion model to advance prior
from time *t* – 1 to time *t*
UAV sensors report time *t* observations;
Exchange sensor info between UAVs
Fuse sensor info based on time *t* prior
using likelihood functions

Figure 5–1: Process flow for Bayesian target estimation using likelihood functions

5.1.1. Defining the target spatial distribution using a Pearson random walk model

The target motion models considered under this investigation are stationary and random walkers. In either case, each target starts with an initial spatial distribution that we will describe later. For the mobile targets, we model their motion as a Pearson random walk. At each time period, the Pearson random walk model involves a target taking a single step with fixed length v in a uniformly random direction $\theta \sim U[0, 2\pi]$.

In Appendix E, we derive several statistical properties for this model. For the Pearson random walk model with step size v, as time approaches infinity, the two-dimensional target motion converges asymptotically to the Bivariate Gaussian Distribution. The Central Limit Theorem can be used to show that the expected distance traveled by the target from its initial position after t time periods is $v\sqrt{t}$ (see Hughes [Hug95] for one such derivation).

One of the technical challenges is discretizing the continuous spatial distribution onto the hexagonal grid used to partition the search space. For a hexagonal cell with side length s whose center is distance $d > 0$ from the center of the distribution, the probability that the target is within that cell after t steps is approximately

$$p = \left[e^{-\frac{1}{2}\left(\frac{d-r_E}{\sigma_t}\right)^2} - e^{-\frac{1}{2}\left(\frac{d+r_E}{\sigma_t}\right)^2} \right] \cdot \frac{3\sqrt{3}}{8\pi d r_E} s^2 , \qquad (5.1)$$

where s and d are defined above and the other variables are as follows:

$$\begin{aligned} \sigma_t &= v\sqrt{\tfrac{1}{2}t} \quad \text{and} \\ r_E &= s\sqrt{\frac{3\sqrt{3}}{2\pi}} \approx 0.91\, s . \end{aligned} \qquad (5.2)$$

After renormalizing the cell probabilities to account for small approximation effects, the final distribution resembles the one shown in Figure 5–2. Each hexagonal cell is color-coded by the probability of a target being in that cell. Red, orange and yellow represent relatively high probability areas, such as near the center of the spatial distribution. Green, cyan and blue represent relatively low probability areas, such as along the periphery of the distribution.

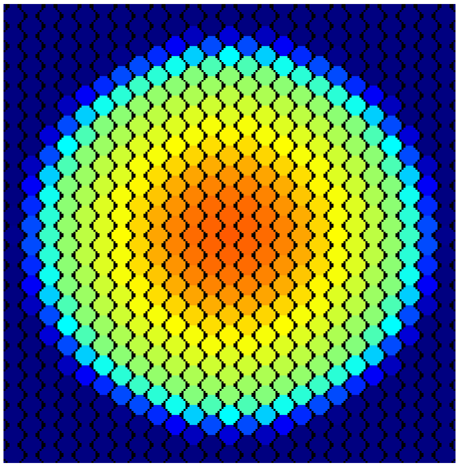


Figure 5–2: Bivariate Gaussian distribution discretized onto hexagonal grid

This random walk model can also be used to specify the initial spatial distribution on the target location. For example, we can generate the initial distribution for a stationary target by applying the distribution associated with 500 random steps of a given length. After generating that initial distribution, the target stops moving for the remainder of the simulation.

We can use a similar methodology to generate multimodal, normal distributions for the initial position. In this case, we select a number of “seed” locations at random uniformly in the search area and then grow a Gaussian distribution based on a number of random steps around these locations. Figure 5–3 shows an example of this type of multimodal spatial prior distribution.

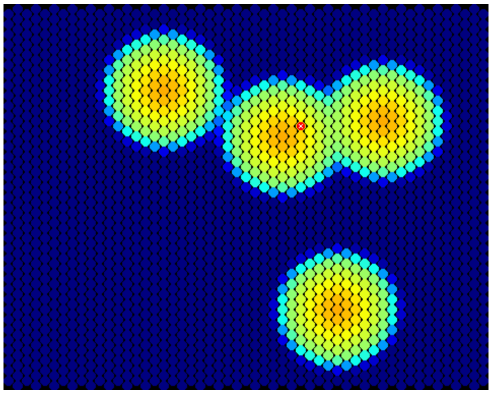


Figure 5–3: Example of a multimodal, Gaussian spatial prior distribution

5.1.2. Defining a motion update for moving targets

The purpose of a motion model is to update the prior distribution given the anticipated motion of each target over time. Define $p_i\,(t–1)$ to be the probability that a target is in cell i at time t–1. The motion update transforms the prior from time t–1 to time t, before any sensor reports have been applied. Functionally, we have

$$p_i^-(t) = \sum_j q\left(\text{target moves to cell } i \text{ at time } t \mid \text{ target is in cell } j \text{ at time } t-1\right) \cdot p_j\left(t-1\right). \quad (5.3)$$

That is, equation (5.3) allocates the new probability weight in cell i at time t based on the probability of targets moving from cell j at time t–1 to cell i at time t. In Appendix E.2, we derive $q(i\,|\,j)$ for the Pearson random walk model with small step size v. Under this derivation, only targets in cell i or its immediate neighbors can step (transition) into cell i during the next time interval.

Consider a target located inside a hexagonal cell with side length s, with the location chosen at random uniformly within the cell. The probability that the target will leave that cell in the next step via a Pearson random step of length v is

$$q^* \; = 1 - \frac{1}{\pi}\left(2\alpha - \sin 2\alpha\right) \text{ where } \alpha = \cos^{-1}\left(\frac{v}{2r_E}\right). \quad (5.4)$$

The variable r_E is defined as in equation (5.2), where it is approximately equal to 0.91 s. If the target leaves its cell in the next step, then it is equally likely to step into any of the six immediate neighboring cells. In addition, the target will remain in the original cell with probability 1–q^*. Thus, the motion update equation can be expressed as

$$p_i^-(t) = \sum_j q\left(i \mid j\right) \cdot p_j\left(t-1\right), \text{ where} \quad \text{(Motion update)} \quad (5.5)$$

$$q(i \mid j) = \begin{cases} 1 - q^* & \text{if } \; j = i \\ q^*/6 & \text{if } \; j \text{ is an immediate neighbor of } i\,. \\ 0 & \text{otherwise} \end{cases}$$

No renormalization is necessary after the motion update because the transition model conserves probability. In addition, note that this derivation takes advantage of the Markov (memory-less) property of the Pearson random walk model. This property states that the probability of a target moving to a particular location in the next time step depends only on the target's current location and not on the history of how the target reached that location.

5.1.3. Specifying the binary sensor model using a likelihood function

As a UAV flies over a cell, we assume that the UAV has a binary sensor model for target detection. That is, at each time step, the sensor either detects a target or it does not. The sensor has a fixed radius of detection, and we assume that each time step yields an independent observation of target presence. In Figure 5–4, we show two UAVs represented by the solid-colored circles. The thin halo surrounding each UAV is the fixed radius sensor footprint, which encompasses the ring of cells neighboring the UAV position.

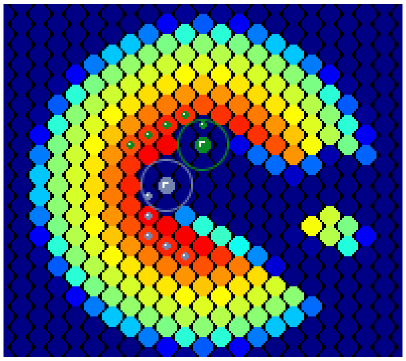


Figure 5–4: Illustration of sensor footprint as a fixed radius halo around each UAV

If the sensor does not detect the presence of a target within the sensor footprint, then a "no target present" report is broadcast to all of the other UAVs for incorporating into each of their probability maps using a sensor likelihood function. Likelihood functions provide a means for updating the probability in each cell based on the observations and reliability of the UAV sensor.

The sensor likelihood function is the likelihood of a sensor observation given a particular ground truth target state (which in this case denotes target position). More precisely, the likelihood function L for the random variable X and observation $Y = y$ is defined to be

$$L(y \mid x) = P\left(Y = y \mid X = x\right) \text{ for } x \in S. \tag{5.6}$$

Note that the likelihood is a function of the target state s, not the observed state y, which is fixed once an observation is made. In our typical usage, $L(y \mid \cdot)$ will not be a probability density function on S.

Using likelihood functions as the starting point for designing an optimal search pattern has several desirable properties. In particular, the use of likelihood provides a common currency that allows the optimal combination of very different kinds of information in a simple manner. A likelihood function enables us to represent sensor information in terms of likelihood which then can be projected onto a surface that describes $L(y \mid \cdot)$ for the entire space of possible target positions $x \in S$.

Metron has extensive experience over the past twenty years using likelihood functions, and an associated technology called likelihood ratio tracking, to fuse sensor information in several successful antisubmarine warfare (ASW) applications, as detailed in Stone, *et al.* [SBC99]. In this context, likelihood functions characterize the output of different systems in order to provide seamless interoperability between systems and sensors.

Figure 5–5 shows an example of how likelihood surfaces from different sensor systems can be multiplied together to produce combined likelihood surfaces for a Naval application. The top left surface shows a bearing likelihood surface for a given set of sensor measurements that suggests that the target position is more likely to be along the diagonal than along the axes. That is, given the observation y, the likelihood function $L(y \mid \cdot)$ is maximized at those target positions s that would be mostly likely to induce the sensor observation y. In this case, the observed sensor data is most likely to have resulted from a target along a particular line of bearing along the diagonal.

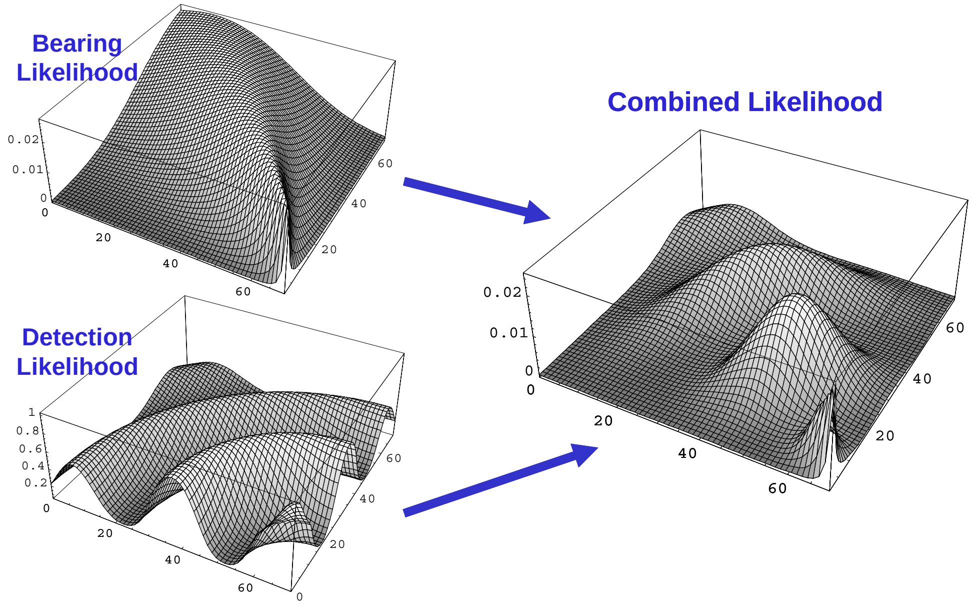


Figure 5–5: Combining bearing and detection likelihood surfaces by multiplication

The bottom left surface in Figure 5–5 is a detection likelihood surface that is high at fixed range intervals and low between those intervals. Pointwise multiplication of these two surfaces for each possible target position yields the combined likelihood surface to the right, which suggests that the target position is most likely to be at fixed intervals along that diagonal (line of bearing).

Returning to the UAV model with a simple binary sensor that we investigated under this research effort, Table 5–1 shows how to convert the binary sensor reliability model into a suitable likelihood function. In this table, p^D is the probability of reporting that a target is present given that a target is present within the sensor footprint (detection probability). The false alarm probability, p^{FA}, is the probability of reporting that a target is present given that a target is not present within the sensor footprint. Each row shows the sensor reliability given the ground truth state, and the probabilities in each row sum to one. The column elements, however, do not necessarily sum to one.

		Reported State	
		Target Present	Target Not Present
Ground Truth State	Target Present	p^D	$1 - p^D$
	Target Not Present	p^{FA}	$1 - p^{FA}$

Table 5–1: Derive likelihood function (columns) from sensor reliability model (rows)

False positive readings, due to a value of $p^{FA} > 0$, require sophisticated data association algorithms in order to suggest which target, if several are present, is the one that has been detected falsely. To limit the scope of this investigation and to focus the research attention on the UAV interactions, we made the simplification that $p^{FA} = 0$, which means that a sensor reports that a target is present only when a target is present within the sensor footprint. Similarly, when a target is not present within the sensor footprint, then the sensor always reports “not present”.

5.1.4. Fusing the sensor information into the motion-updated prior distribution

The final step in the process is to update the target prior distribution using the sensor-based likelihood function to produce the time t posterior distribution. This calculation uses the motion-updated prior $p^{-}(t)$ from equation (5.5) and the likelihood function from the previous section. If the target is assumed stationary, then $p^{-}(t) = p(t\text{–}1)$. The update has the functional form,

$$p_i(t) \ = \ \frac{1}{C} L(y_t \,|\, i) \cdot p_i^{-}(t) \quad \text{(Information update)}\,. \tag{5.7}$$

The variable C is a renormalization factor that ensures that $\sum_i p_i(t) = 1$ after the information update is complete.

With respect to implementing this information update, there are a few subtleties to address when multiple targets are present. Each UAV sensor reports either “target present” or “target not present” for the set of cells within its sensor footprint. If there are multiple targets in the simulation, then we assume that there is an independent sensor report for the state of each target. That is, if there are three targets, then UAV1 may report that target 1 is not present, target 2 is not present and target 3 is present within its sensor footprint. The UAVs maintain probability estimates for each target as a separate layer, with layer n corresponding to target n. The sensor report for target n is fused into layer n, independent of the other observations.

Table 5–2 shows the explicit update for target n based on the likelihood function associated with cell i. In this case, both $p_{in}(t)$ and $p^{-}{}_{in}(t)$ are distributions for target layer n. The normalization factor C_n is also specific to target layer n.

	Reported State	
	Target n Present	Target n Not Present
Cell i inside Footprint	$p_{in}(t) = \frac{1}{C_n} p^D \cdot p_{in}^{-}(t)$	$p_{in}(t) = \frac{1}{C_n}\left(1 - p^D\right) \cdot p_{in}^{-}(t)$
Cell i outside Footprint	$p_{in}(t) = 0$	$p_{in}(t) = \frac{1}{C_n} p_{in}^{-}(t)$

Table 5–2: Convert prior distribution into posterior using sensor likelihood function

Having illustrated the mechanics of performing the Bayesian updates (both motion and sensor information), we proceed to the search path optimization algorithms that were developed for the UAVs.

5.2. FINITE-HORIZON SEARCH PATH PLANNING

In this section, we describe a finite-horizon approach by which each aircraft optimizes a search path that maximizes the probability of detecting a target over that horizon. Finite-horizon planning has several attractive features. In particular, it focuses attention on optimizing collectively the probability of detection given real-time spatial distributions, and UAVs can continue to optimize locally even when communications are disrupted.

Using finite-horizon planning and sharing sensor information as described in the previous section, UAVs do not need to share their entire tactical picture (target probability maps) or even maintain identical tactical pictures (loss of synchrony) across UAVs. As bandwidth constraints tighten, the system performance will degrade because UAVs will not be acting with full, timely information, but it will degrade gracefully when bandwidth requirements are low and quickly improve when bandwidth becomes more plentiful and UAVs can share past sensor reports. These properties are desirable in terms of designing effective, robust system interaction mechanisms.

However, there can be significant computational costs associated with relatively long planning horizons. In addition, there is no guarantee that the union of search plans for the fleet of UAVs will be effective jointly because the search paths are optimized independently for each UAV. We will discuss approaches later in this chapter for handling both of these issues through the use of a genetic algorithm to reduce the exponential explosion of possible search paths and a deconfliction algorithm to reduce redundant effort in the joint search plans.

5.2.1. Discounted finite-horizon search path planning

We have designed a rolling-horizon planning approach in which each UAV constructs a path that maximizes the probability of target detection over a fixed horizon of length *T*. At each step in the path, a UAV must choose to bank left, bank right or go straight. Each of these movement choices moves the UAV to one of three adjacent cells. In Figure 5–6, a series of small circles extending from each UAV illustrates the optimized five-step search path associated with that UAV.

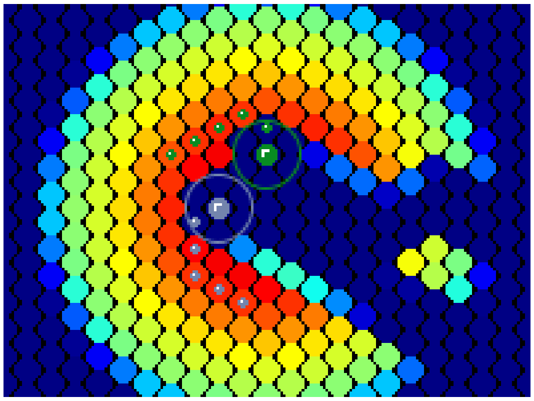


Figure 5–6: Two UAVs optimize five-step look-ahead search paths

As before, let $p_{in}(t)$ be the prior distribution that defines the probability that target n is present in cell i at time t. A T-step look-ahead that maximizes the probability of detecting at least one target means that the UAV must choose a path of cells $\boldsymbol{i} = \{i(1), i(2), \ldots, i(T)\}$ that optimizes the value $R_i(t, T, \boldsymbol{i})$ over that search path,

$$R_i(t,T,\boldsymbol{i}) \;=\; 1-\prod_{s=1}^{T}\left[\prod_{n=1}^{N}\left(1-p^D\cdot p_{i(s),n}(t+s)\right)\right] \approx \sum_{s=1}^{T}\sum_{n=1}^{N} p^D\cdot p_{i(s),n}(t+s)\,. \qquad (5.8)$$

There are 3^T feasible paths from which to choose, so the optimization problem grows exponentially as T increases. The first equality in equation (5.8) specifies the probability of detection at least one of the N targets over the T periods, and the second equality specifies an approximation (the expected number of targets detected over the path $\boldsymbol{i}$) that is computationally simpler and leads to similar rank ordering of the 3^T paths.

This approximation has another benefit. When optimizing a search path over a finite horizon, each step in that path has the same influence on the objective function. However, the first step in the path is more important to optimize than the fifth step, especially when reoptimizing after each time period, because the planning information will change over the first four steps. For example, planning to visit a particular cell at time t+5 may be rendered less effective later on because another UAV visited that same cell at time t+2.

Consequently, we will use a discount factor $0 < \lambda < 1$ to reduce the influence of the later cells in the path. This discount factor can be incorporated easily into the approximation in equation (5.8) as follows,

$$\begin{aligned} R_i(t,T,\boldsymbol{i},\lambda) &= \sum_{s=1}^{T}\sum_{n=1}^{N}\lambda^{s-1}\left(p^D \cdot p_{i(s),n}(t+s)\right) \\ &= p^D\sum_{s=1}^{T}\lambda^{s-1}\cdot\left(\sum_{n=1}^{N}p_{i(s),n}(t+s)\right). \\ &= p^D\sum_{s=1}^{T}\lambda^{s-1}\cdot p_{i(s)}(t+s) \end{aligned} \tag{5.9}$$

This approach gives full value to the expected number of detections at time t+1, applies a factor λ to the expected number of detections at time t+2, applies a factor λ^2 to the expected number of detections at time t+3, and so on. When λ is close to one, there is little change to the objective function. However, as λ decreases, the influence of cells in the early part of the path starts to dominate that of cells later in the path.

5.2.2. Deconfliction and other implementation details

There are several implementation details that must be addressed. First, when a UAV travels along its search path, the sensor reports reflect any targets that may be within the sensor footprint, not just within the cell in which the UAV is located. The objective function in equation (5.9) assumes single-cell probabilities. In practice, the value associated with $p_{i(s)}(t+s)$ must contain all of the probability within the sensor footprint centered on cell $i(s)$. Under this study, the footprint contains cell $i(s)$ and its immediate six neighbors.

Second, when a UAV receives a “no target present” sensor reading centered on cell $i(1)$ and then moves to cell $i(2)$, then the value associated with being at cell $i(2)$ must incorporate the “no target present” reading at cell $i(1)$ in the previous time period. When computing the expected number of detections along a path, that path score must consider the impact of sensor measurements along the way. We will assume in our calculation of path scores that the value at cell $i(s)$ is conditioned upon fusing “no target present” readings when the UAV is located at cells $i(1)$, …, $i(s–1)$. Without this conditioning, the optimal search path may simply orbit a high-valued cell. Although this complicates the calculation of path scores, it is necessary to improve the search performance.

Next, we address the need for UAVs to deconflict their search paths. Occasionally, two UAVs will choose similar paths in which one UAV travels along the same path but behind another UAV. This is common because each UAV attempts to ascend the steepest available gradient, and two nearby UAVs may be attracted to the same gradient.

Deconfliction is a process by which UAVs share search path information and reoptimize their paths based on the paths chosen by the other UAVs. When two UAVs share similar paths, deconfliction tends to cause one UAV to follow a parallel gradient that is off to the side of the other search path.

Figure 5–6 cited earlier shows this type of effect. The two UAVs enter the prior distribution side-by-side. In the illustration, the blue-grey UAV is about to make a sharp turn to its left while the green UAV is making a more gradual turn that falls outside the blue-grey UAV’s path. The result is better search coverage and less redundant search effort.

UAVs perform deconfliction by sharing their proposed search paths, each of which is optimized solely by maximizing the expected detections along a path and not taking into account the proposed search paths of the other UAVs. Once those optimized paths are published and shared across the set of UAVs, each UAV is given the opportunity to change its decision based on what the other UAVs propose to do.

In order to limit the amount of negotiation necessary to get the paths to converge, we select one UAV at random (or based on a rank ordering of the UAV indices). That UAV is given the opportunity to reoptimize its search path with respect to the paths proposed by the other UAVs. The new search path for that UAV is then fixed, and then the next UAV gets an opportunity to reoptimize. This reoptimization process occurs for all UAVs once, although additional rounds could be included if desired. Once all of the search paths have been fixed, each UAV takes the first step in its search path, and the Bayesian update cycle illustrated in Figure 5–1 continues.

5.3. GENETIC ALGORITHM IMPLEMENTATION

As noted earlier, the possible number of paths grows exponentially in the path length T. Rather than enumerate and evaluate all 3^T paths to select the path $\boldsymbol{i}^*$ that maximizes $R_i(t, T, \boldsymbol{i})$, we developed a customized genetic algorithm that prunes the search path space to find a high-quality search path quickly. As we will show later in the experiments section, the genetic algorithm worked well empirically because the path planning is driven by hill-climbing on the probability distributions, and genetic algorithms are well-suited for this type of optimization. Below, we provide the details of the algorithm that we developed.

Genetic Algorithms (GAs) are a directed search or evolutionary optimization tool used to evolve a population of candidate solutions to a given problem using operations inspired by natural genetic variation and natural selection (see, for example, [Gol89] or [Mit96]). GAs are useful when a solution space is relatively large and smooth; i.e. when all solutions in a neighborhood have roughly the same solution quality. GAs evolve populations of chromosomes (search paths) that consist of sequences of genes (individual path steps) in which each gene is chosen from a set of alleles (the set of possible path steps).

The GA implementation of the target search problem starts with an initial population of randomly generated search paths of length T and performs the selection, crossover, and mutation operations on the population to breed successive generations of new path populations. The probability of detecting a target along the path serves as the "fitness function" for each of the paths. In this approach, a population of L chromosomes breeds L new chromosomes. The GA preserves the best (most "fit") L chromosomes from the $2L$ available chromosomes for the next generation of selection, crossover and mutation.

5.3.1. Search path encoding

We considered two different approaches for encoding search paths into GA chromosomes: (1) using an absolute path orientation and (2) using a relative path orientation. Both of these representations are illustrated in Figure 5–7 and begin with a UAV in the center cell with a particular orientation. Due to banking constraints, we will assume that at each step the UAV is limited to moving to one of three adjacent neighbors relative to its orientation, either left, right or straight.

In the absolute orientation, the set of alleles map an *absolute* direction to each of the adjacent neighboring cells. The six alleles are assigned unique labels based on the cardinal points on a magnetic compass: north (N), northwest (NW), southwest (SW), south (S), southeast (SE), and northeast (NE). In Figure 5–7, the sample path, or chromosome, using the absolute orientation is encoded as {NE, N, N, NW, N, NE}.

In the relative orientation, the set of alleles map a *relative* direction to each of the three neighboring cells into which the UAV can move feasibly based on the banking constraints. The alleles are assigned labels based on these three feasible directions: left (L), straight (S) or right (R). In Figure 5–7, the chromosome using the relative orientation is encoded as {S, L, S, L, R, R}.

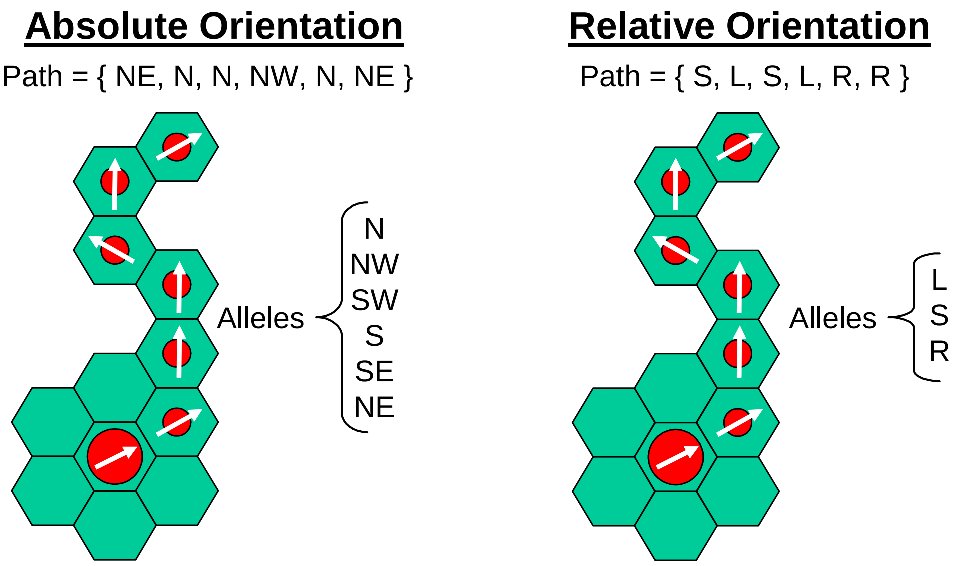


Figure 5–7: Encoding scheme for describing the path chromosome

Each of these encoding schemes has advantages and disadvantages relative to each other. One advantage of the absolute path encoding scheme is that the crossover and mutation operations that we describe later tend to breed new chromosomes that preserve the direction of high valued future path steps. For example, if there is a high valued region to the north, then GA chromosomes with genes containing the "north" allele would tend to have a higher fitness then other chromosomes. These chromosomes in turn would have a greater chance of breeding and spreading their beneficial genes.

However, the primary disadvantage of using the absolute path encoding approach is that the crossover and mutation operations may evolve paths, such as {N, SE, SW, NE}, that are not feasible with respect to the UAV banking constraints. This is an important issue because, for a path of length T, there are 6^T possible encoded paths, but only 3^T of those paths are feasible. For example, if $T = 5$, then only three percent ($1/2^5$, or 1/32) of the possible paths are feasible.

For the relative path encoding, the crossover and mutation operations are guaranteed to evolve feasible search paths because the genes encode only feasible movements (L, R, S). The main disadvantage of the relative encoding is that the crossover and mutation operations do not preserve the direction of the tails of the paths. For example, consider the example in Figure 5–8. One small change in one of the genes from Left to Right leads to a large change in the cells visited at the end of that path.

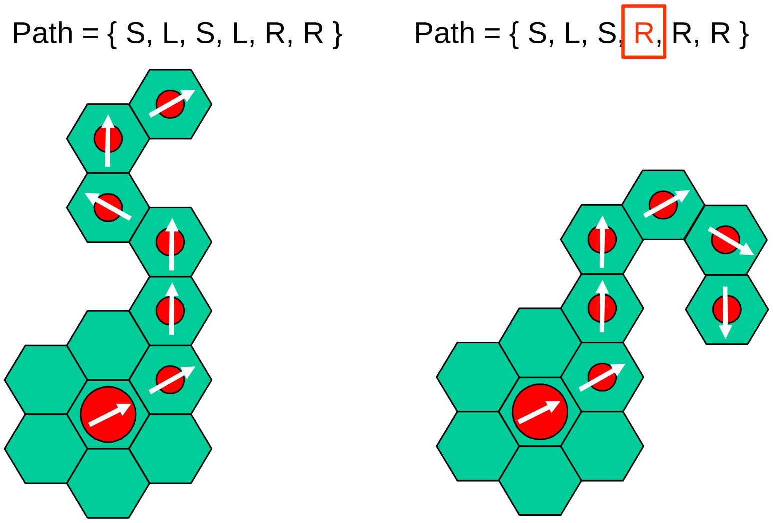


Figure 5–8: Relative encoding approach does not preserve direction for future moves

Although the absolute path encoding preserves the tails of highly fit paths, we choose to use the relative path encoding instead to ensure that all evolved chromosomes produce feasible search paths.

5.3.2. Breeding operations for next generation (selection, crossover and mutation)

To perform the GA heuristic on a population of chromosomes, a pair of parents is selected (with replacement) to create two offspring through Crossover. The expected number of times a chromosome is selected for breeding is proportional to the fitness of the chromosome, which we define as the sum of the target probabilities along the search path described by the chromosome. For a population of size *L*, each generation of the GA attempts to perform *L* crossovers.

The crossover operation is performed on a pair of parents with probability $p^c \approx 1$. If the operation is performed, then a crossover point *X* is chosen at random and the parent chromosomes create two offspring (see the example in Figure 5–9). If crossover is not performed, then the parent chromosomes are cloned exactly.

The crossover operation leads to two offspring. The first offspring has the first *X* genes from the first parent and the last *T*–*X* genes from the second parent. The second offspring has the first *X* genes from the second parent and the last *T*–*X* genes from the first parent. The fitness score for each offspring is computed and stored.

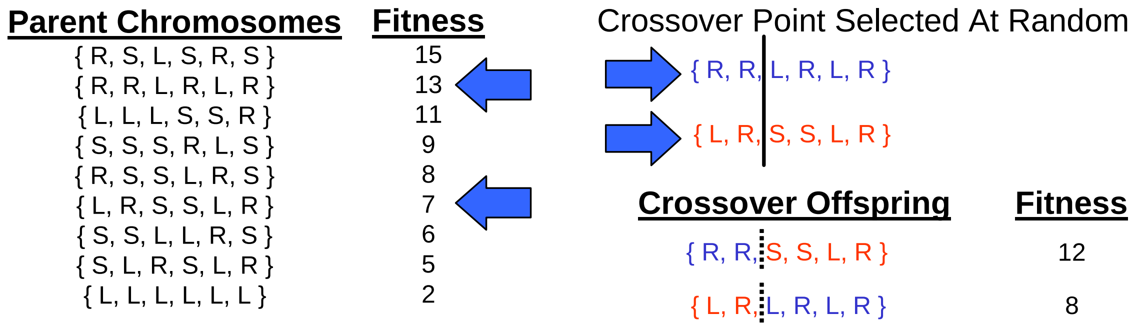


Figure 5–9: Crossover operation used to update the path population

The mutation operation is performed on each gene in the new offspring chromosomes with probability $p^m << 1$. However, if crossover was not performed (i.e., the offspring are clones of the parent chromosomes), then the mutation operation is performed on each gene in the new offspring with probability $p^m < 1/3$ to increase the population diversity.

When performing mutations, the algorithm cycles through all the genes in a chromosome and makes an independent random draw to determine if that gene should be mutated. If mutation is required, then the current allele at that gene (either L, R or S) is replaced with one of the other two alleles with equal probability.

At the end of a new generation of chromosomes, the GA compares the fitness scores of all of the parent and offspring chromosomes, and keeps the L chromosomes with the highest fitness score to be the next generation. At the end of the last generation, the GA heuristic executes the search path with the highest chromosome fitness score.

5.3.3. Selection of the population size given a planning horizon length

One of the challenges of designing a genetic algorithm is to determine an appropriate population size L and the number of generations G to evolve. Obviously, if the product $L \cdot G$ exceeds the enumeration of all possible search paths 3^T, then the genetic algorithm does not save any computational effort. In particular, we want to find a functional form for that product that scales polynomially rather than exponentially.

We performed a number of informal experiments to determine appropriate forms for L and G, the details of which we do not include here. Given a path length T, the goal was to find the smallest population and number of generations that provides similar search performance to using the best of the 3^T possible paths. We found that the following forms performed well,

$$G = \tfrac{1}{4}T^2 \tag{5.10}$$

$$L = 5G = \tfrac{5}{4}T^2 . \tag{5.11}$$

In practice, we round G up to the nearest integer and round L up to the nearest even integer so that the number of potential parents is an even number. The total number of paths considered is then approximately $LG \approx \tfrac{5}{16}T^4$, which does not grow exponentially as does the enumerative approach.

One of our observations is that having a constant for the ratio L/G led to good solutions. Consider the case where we fix the product $L \cdot G$ and apply extreme forms for L and G. If $G = 1$, then the number of paths considered is L, which means that the GA will pick the best of the randomly generated, initial generation of paths. In the other extreme, if $L = 1$, then the GA will evolve a single path for G generations, replacing the child with the parent only if the child has a higher fitness score. Balancing L and G to have a constant ratio provides the right trade-off between randomly generating paths and evolving successful paths.

Table 5–3 compares the growth of the number of paths used by the genetic algorithm and by the enumerative approach. For small values of T, the total number of paths is similar, but as T increases, so does the gap between the number of paths for each approach.

T	L	G	GA PATHS ($L \cdot G$)	ENUMERATIVE PATHS (3^T)
4	20	4	80	81
5	32	7	224	243
6	46	9	414	729
7	62	13	806	2187
8	80	16	1280	6561
9	102	21	2142	19683
10	126	25	3150	59049

Table 5–3: Number of paths needed for similar search performance by enumerative and genetic algorithm approaches

5.4. Experimental Design and Results for UAV Search

In this section, we present results from a series of experiments performed to evaluate the performance of the finite-horizon search planning approach described in this chapter. There are three classes of experiments, all involving the coordinated search for a single, stationary target. The results for multiple or mobile targets are similar.

For all of the experiments in this section, we assume a search area with dimensions 498 units in width by 499 units in length (55 hexagonal cell columns by 95 hexagonal cell rows). Each of the experiments uses values selected from the following parameters:

- Number of UAVs, M;
- Target prior distribution information (σ and number of modes, k);
- Probability of detection, p^D for UAV sensors;
- Length of planning horizon T used by each UAV; and
- Path optimization using enumeration or the genetic algorithm.

The prior distribution on target location is based on a Pearson random walk model, the properties of which are described in Appendix E. Given a target following a random walk with fixed step size v for t time steps, the resulting prior is a Gaussian distribution with $\sigma = v\sqrt{\frac{1}{2}t}$. In some cases, we use a multi-modal Gaussian distribution with k modes, each of which has $\sigma_k = \sigma / \sqrt{k}$. This is equivalent to splitting the t random walk steps across the k modes equally. To aid the reader's intuition regarding the relative search difficulty for different priors, Figure 5–10 illustrates typical prior distributions for different combinations of k and σ that will be used in the experiments.

The evaluation will focus on operational metrics, primarily the median time needed for one of the UAVs to detect the target, based on statistics collected over hundreds or thousands of independent trials. To investigate tail effects, we will supplement the median statistics with the 25th and 75th percentiles on the time to target detection. In the next chapter, which covers search model extensions, we will also consider target containment statistics, such as the average or root-mean-squared (RMS) error between the estimated target location and the actual location over time.

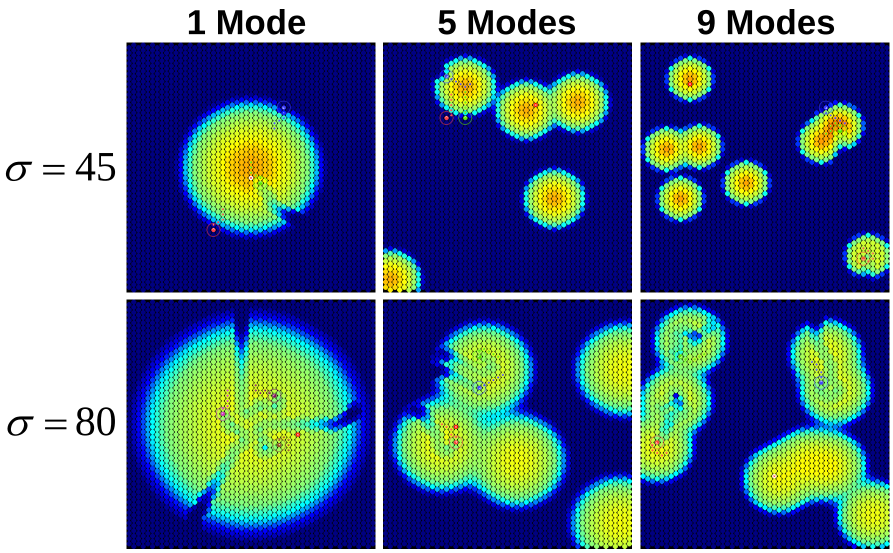


Figure 5–10: Target prior distributions based on number of modes and diffusion

In section 5.4.1, we compare the results of UAVs sharing or not sharing sensor information and with or without explicit path deconfliction. In section 5.4.2, we compare the search performance using an enumerative path planning approach with the genetic algorithm designed in section 5.3. Finally, in section 5.4.3, we compare the operational performance of the path planning-based coordinated search against a pre-defined lawnmower search pattern.

5.4.1. Value of sensor information sharing and deconfliction

In this section, we compare the results of sharing or not sharing sensor information and with or without explicit path deconfliction for three UAVs searching for a single, stationary target using the finite-horizon planning approach.

Experimental Design. Table 5–4 summarizes the simulation parameter settings for the finite-horizon planning experiments with a single, stationary target. The initial prior distribution has a single mode with σ = 47 and is centered within the search area square. The three search UAVs are placed on the perimeter of a circle centered on the prior with radius approximately equal to 3.7σ. We consider cases in which the UAVs have initial angular separation on the circle of θ = 10°, θ = 60°, or θ = 120°.

The UAVs use a five-step planning horizon in which all search paths of length five are considered (an "enumerative" approach). We assume a perfect UAV sensor, i.e., p^D = 1. We compare the time to target detection results with and without UAV path deconfliction as well as with and without sharing UAV path history.

PARAMETERS	VALUES
Number of UAVs	$M = 3$
Target Prior (σ and # of modes)	$\sigma = 47$; single mode
Probability of Detection, p^D	$p^D = 1$
Path planning horizon	$T = 5$
Path planning algorithm	Enumerative

Table 5–4: Parameter settings for information sharing and deconfliction experiments

Experimental Results. For each initial angular separation value θ and each combination of deconfliction / information sharing, we record the target detection time for 5,000 independent trials. Figure 5–11 shows the median, 25th and 75th percentiles for the time to target detection for each approach. The results show that sharing sensor information history significantly improves search performance. Deconfliction is less important, especially when sensor data is shared. When not sharing history, the angular separation has a greater influence on the results, especially when UAVs with a small angular separation (i.e., UAVs that start close to each other) do not deconflict paths. In subsequent experiments, we will assume that UAVs share sensor information and deconflict search paths.

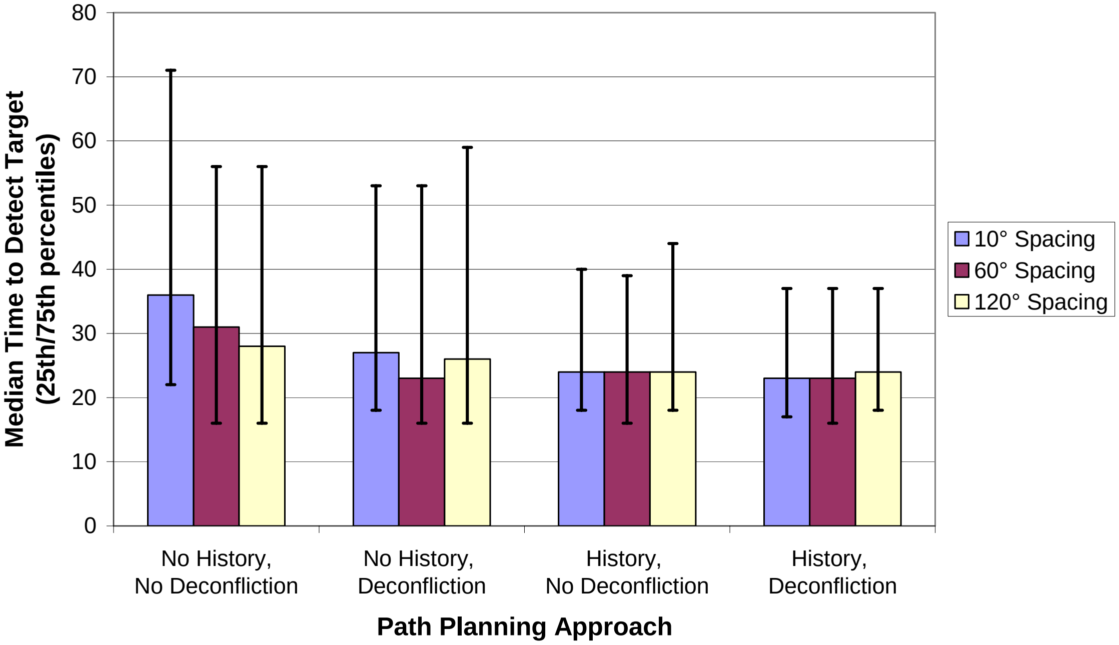


Figure 5–11: Results of information sharing and deconfliction for search planning

5.4.2. *T*-Step finite horizon planning (enumerative versus genetic algorithm)

In this section, we compare the search performance as a function of planning horizon length T for three UAVs searching for a single, stationary target using either the enumerative approach or the genetic algorithm.

Experimental Design. Table 5–5 summarizes the parameter settings for the planning horizon experiments. There are $k = 9$ modes in the initial prior distribution with a standard deviation of $\sigma = 45$ units split between the modes (i.e., each mode has $\sigma_k \approx 15$ units).

The nine modes are distributed at random uniformly throughout the search area. Recall from earlier in the section that Figure 5–10 illustrates instances of the initial prior distribution for different number of modes and standard deviations. The initial locations of the three UAVs are also randomly drawn from a uniform distribution.

For the enumerative case, we consider planning horizon lengths of $T = 1, 3, 5$, and 7. For the genetic algorithm, we consider only the cases $T =7$ and 9. We assume that the UAVs have imperfect sensors, i.e., $p^D = 0.4$.

PARAMETERS	**VALUES**
Number of UAVs	$M = 3$
Target Prior (σ and # of modes)	$\sigma = 45$; 9 modes ($\sigma_k = 15$ for each mode)
Probability of Detection, p^D	$p^D = 0.4$
Path planning horizon	$T = 1, 3, 5, 7, 9$
Path planning algorithm	Enumerative or Genetic algorithm

Table 5–5: Parameter settings for finite-horizon planning experiments

Experimental Results. For each planning horizon length T and planning algorithm, we record the target detection time for 3,500 independent trials for $T < 9$ and 1,000 independent trials for $T = 9$. Figure 5–12 shows the median, 25th and 75th percentiles for the time to target detection for each approach. The number above the curves is the number of paths considered for the enumerative approach (all possible paths), and below the curves is the number of paths considered for the genetic algorithm.

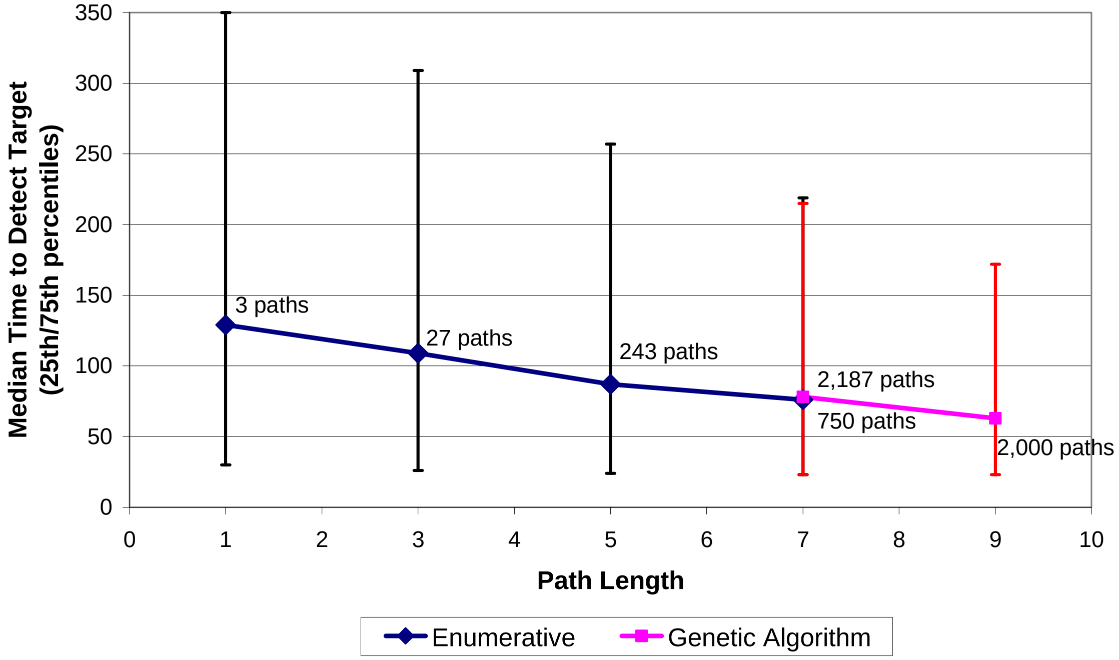


Figure 5–12: Comparison of enumerative and genetic algorithm search performance

The detection time steadily decreases as the planning horizon lengthens. However, for a path length $T = 7$, the genetic algorithm is able to match the performance of the enumerative algorithm with one-third the number of paths. Moving to a path length $T = 9$, the genetic algorithm is able to improve the search performance even further using the same number of paths as enumerative with path length $T = 7$. In addition, for a path length $T = 9$, the genetic algorithm requires an order of magnitude fewer paths than enumerative would require (19,683 possible paths).

5.4.3. Comparison against lawnmower search pattern

Finally, we compare the search performance as a function of the number of UAVs searching for a single, stationary target using either a five-step enumerative approach or a lawnmower search pattern that we will describe.

Experimental Design. Table 5–6 summarizes the parameter settings for the variable number of UAVs experiments. There is a single Gaussian mode centered in the search space with a standard deviation of $\sigma = 80$ units. The initial locations of the $M = 3, 5, 7$ or 10 UAVs are equidistant around a circle with fixed radius.

PARAMETERS	**VALUES**
Number of UAVs	$M = 3, 5, 7, 10$
Target Prior (σ and # of modes)	$\sigma = 80$; single mode, centered in search area
Probability of Detection, p^D	$p^D = 0.2$
Path planning horizon	$T = 5$
Path planning algorithm	Enumerative or Lawnmower

Table 5–6: Parameter settings for variable number of UAVs experiments

The Lawnmower search technique provides coverage over an entire search area in a manner similar to mowing a lawn (see Figure 5–13). For the case of a single UAV, the UAV starts in the upper-left corner of the search area and travels down until it reaches the opposite edge. The UAV then banks and reverses direction, traveling up along a path adjacent and parallel to the previous downward path. This process continues until the UAV reaches the right side of the search area, at which time the UAV travels along the top edge to the original starting point and begins again.

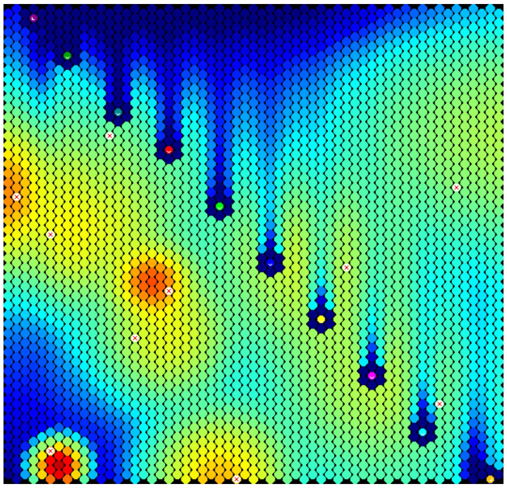


Figure 5–13: Illustration of ten UAVs following Lawnmower pattern

The important point is that the UAV visits each cell once on a closed loop cycle through all cells in the search area before repeating the pattern. When multiple UAVs are available, they can be spaced equidistant to each other and move together along the closed cycle, as shown in the figure. Using multiple UAVs in this manner increases the frequency of cell visits along the repeated lawnmower path.

The Lawnmower search pattern is a centralized approach that is very efficient at covering the entire search area in the minimal amount of time. In particular, this pattern provides optimal detection when there is no known distribution for the target location (i.e., assume a uniform spatial prior distribution). However, when a non-uniform prior is available, the Lawnmower searches fail to exploit this probabilistic structure.

Experimental Results. For different numbers of UAVs, we record the target detection time for 1,500 independent trials for the five-step enumerative and lawnmower planners. Figure 5–14 shows the median time to target detection for each approach, and the 25th and 75th percentiles for the enumerative approach.

Both approaches improve the detection time given additional UAVs. However, the enumerative approach detects the target in roughly one-third the time needed by the lawnmower searchers. Another way of interpreting the two curves is that three UAVs following the five-step enumerative approach can detect the target as quickly as ten UAVs following the lawnmower pattern. This shows the value of exploiting the target prior information using the Bayesian nonlinear tracking methodology.

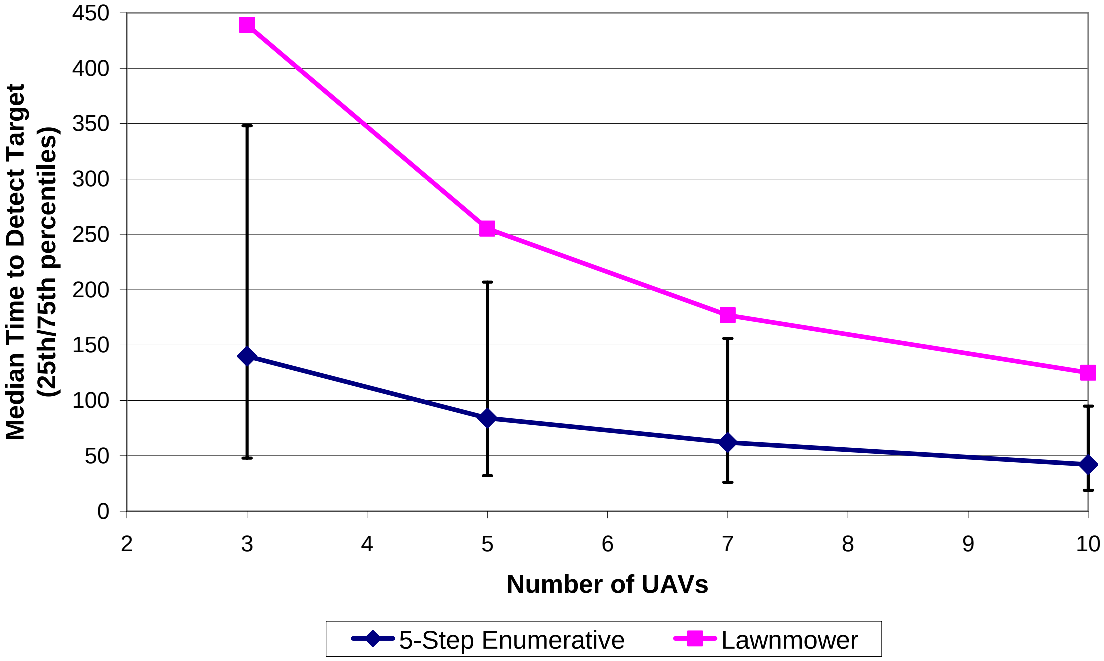


Figure 5–14: Comparison of five-step enumerative with lawnmower for different fleet sizes

5.5. Limitations of finite-horizon planning

In this chapter, we have described the pros and cons associated with the choice of planning horizon length. Longer horizons lead generally to better search plans, but the computational effort grows exponentially, which can be mitigated somewhat through the use of the genetic algorithm that we designed. However, there is another instance in which short horizons (myopic planning) can cause problems. The finite horizon approach relies on finding a probability gradient. However, if no such gradient exists in the vicinity of the UAV, then the UAV may wander aimlessly until it stumbles upon a more productive region of the search area.

Figure 5–15 shows an example of three UAVs that fail to lock into effective gradients that lead to the high probability region. Two-dimensional Gaussian distributions have probability weights that fall quickly with distance, and in this case, all three UAVs are sufficiently far away from the prior that a five-step look-ahead fails to encounter any cells with a non-trivial amount of probability.

This is quite frustrating because all three UAVs are fully aware of the prior distribution, and the target probability maps used in the optimization are exactly as shown in the figure. The problem is that the five-step look-ahead induces a blind spot in the planner to any cells outside the five-cell radius. Ironically, moving targets can make this phenomenon less of a problem because as time advances, the distribution diffuses and eventually the edge of the distribution will reach each of the UAVs.

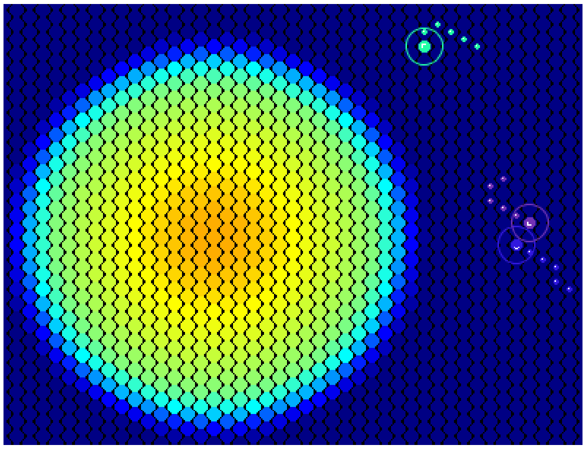


Figure 5–15: Poor search paths resulting from a lack of productive gradients

There are several ways to mitigate this problem. One could lengthen the planning horizon, but that has severe computational consequences and may not solve the problem. For example, the purple UAV in Figure 5–15 would need a planning horizon of at least nine cells to have any probability in its optimal path.

Another approach would default to a simple rule such as “If the optimal path score is less than *X*, then have the UAV move in the direction of the cell with maximum probability in the entire search area.” Determining an appropriate threshold value *X* may require some *ad hoc* tuning to find the right balance between moving along the optimal path and moving to the highest probability cell. In addition, if there are two targets, one with the prior shown in Figure 5–15 and the other with a uniform distribution for the prior, then this threshold approach may not be effective. The reason is that all of the cells will have a non-trivial weight due to the uniform distribution, yet a gradient may not exist.

Rather than developing ways to workaround the finite horizon planning and deciding when to apply this workaround, we developed an approach under a DARPA DSO seedling effort that approximates the results of an infinite-horizon plan and eliminates the concern about finding suitable gradients. A brief summary of this transition of our TASK research is described in section 7.1 and presented in full detail in [God05].

In the next chapter, we describe several extensions to the UAV search and surveillance technologies that we developed and evaluated under the DARPA TASK contract.

6. EXTENSIONS TO UAV SEARCH AND SURVEILLANCE

In this chapter, we address three extensions to the basic search and surveillance technology. First, we integrate a network of unattended ground sensors (UGS) into the search problem, and demonstrate how UAVs can choose collaboratively when to deploy a UGS to minimize search effort. Second, we consider the effects of evasive targets that move partly in response to the UAV locations. Finally, we consider a joint search and surveillance problem. The surveillance UAVs maintain target location estimates while the search UAVs detect targets with uncertain locations. The joint problem involves each UAV deciding dynamically whether to perform a search or surveillance role depending on the marginal value of each task at a given time. We show the effectiveness of our approach in a series of experiments over a wide range of environmental settings.

6.1. UNATTENDED GROUND SENSOR (UGS) NETWORKS

Under this extension, we add a network of deployable, unattended ground sensors (UGS) to augment the aerial search and surveillance performed by the UAVs. Incorporating the UGS network requires elements of coordination, adaptation and resource management, which is consistent with the TASK goals.

The UGS network has several operational characteristics that fit well with the target search and surveillance model:

- UGS networks can be deployed in high-threat areas in which UAVs may be vulnerable;
- UGS networks can be deployed in areas that require more constant monitoring than can be achieved by UAVs; and

- UGS networks can be deployed in areas in which UAV sensors may be less effective due to terrain or other environmental factors.

With respect to the TASK objectives, allowing UAVs to have a limited number of on-board, deployable UGSs that can be air-dropped into an area of interest presents some interesting decisions for the system of UAVs:

- UAVs must consider the value of deploying a UGS now versus the expected benefit of holding the UGS in reserve for later deployment given future uncertainty;

- UAVs must coordinate the UGS deployment (or non-deployment) with each other to reduce redundancy; and

- UAVs must fuse the information provided by the UGS network when updating the target search maps (i.e., how does a search UAV effectively use the stationary UGS network when planning a target search path?).

Figure 6–1 shows two UGSs performing a target detection sweep. The sensor has a sweep angle, and any target within that angle is detected (with some allowable false negative rate). There is a corresponding likelihood function that is used to perform the updates within this sweep angle. As the sensor sweeps the area, the detections or non-detections are shared with the UAVs and cause the target prior to deform. The UAVs then base their motion upon this modified target prior.

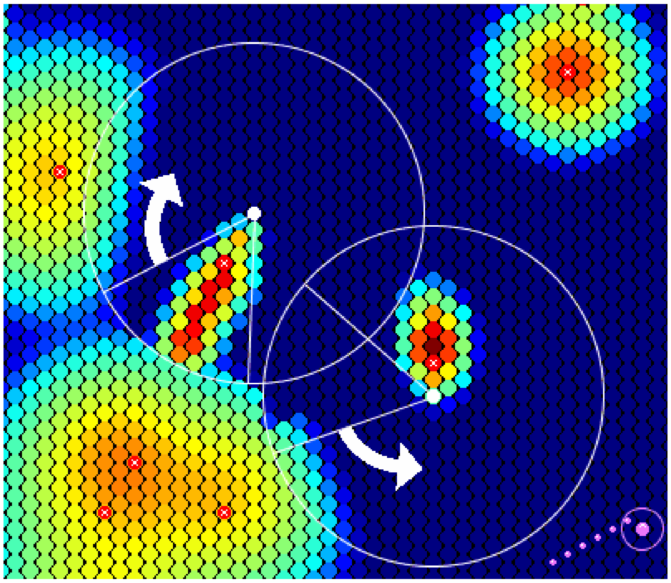


Figure 6–1: Example of two UGSs performing a sweep

6.1.1. UGS deployment approach (fixed threshold)

In order to deploy UGSs effectively, we needed to develop a scheme for scoring the value of deploying a single UGS at a given time and location, and then develop thresholds for deciding at what value a UGS should be deployed. To do so, we used an entropy-based calculation. Figure 6–2 shows a UAV surrounded by a large white circle, which indicates the footprint associated with a potential UGS that could be deployed at that location.

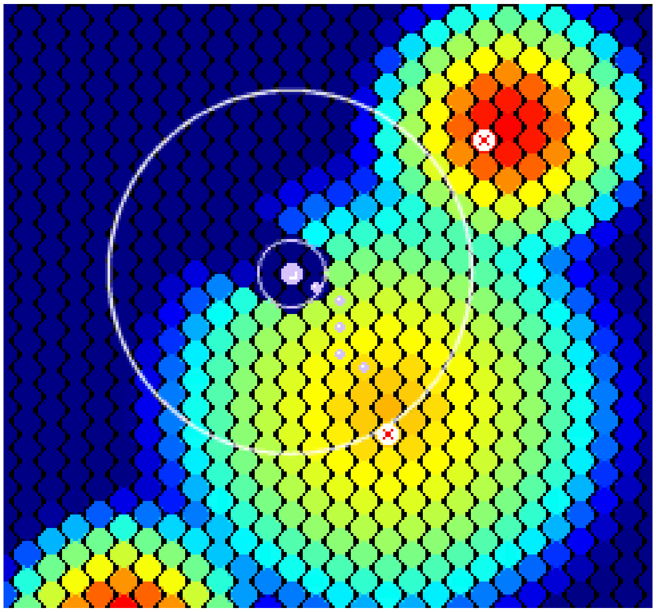


Figure 6–2: UGS footprint used to determine set of cells used for entropy calculation

Given a set of targets, indexed by $n = 1, \ldots, N$, and a set of cells I_m that are contained within the potential UGS footprint of UAV m, we can compute the entropy ξ_m associated with the cells in that footprint as

$$\xi_m = -\sum_{n=1}^{N} \sum_{i \in I_m} p_{in} \cdot \log_2 \left(p_{in} \right). \tag{6.1}$$

This entropy calculation tracks closely to potential deployment value. Deployment is least valuable when there are no targets in the region (i.e., $p_{in} = 0$ for all i, n) or when any target n that is in that region has no uncertainty about location (i.e., $p_{in} = 0$ for all cells but one, and in that cell, $p_{in} = 1$). In either case, each cell probability is zero or one, which means that the entropy is zero (i.e., $0 \cdot \log_2 0 = 0$ or $1 \cdot \log_2 1 = 0$). Thus, the entropy is zero when deployment is least valuable.

Deployment is most valuable (maximum entropy) when the targets have all of its weight inside the UGS footprint and that weight is spread uniformly throughout the region. For one target, this means that the weight in each of the I cells is $1/I$,

$$\text{Max entropy for one target} \;=\; -\sum_{i \in I_m} \left(\frac{1}{I}\right) \cdot \log_2 \left(\frac{1}{I}\right) \;=\; -I \cdot \left(\frac{1}{I} \cdot \left(\log_2 1 - \log_2 I\right)\right) \;=\; \log_2 I \;.$$

The maximum entropy of N targets, then, is $N \cdot \log_2 I$. We can rescale the entropy measurement by this maximum value. By doing so, we can express the threshold for deciding when to deploy as a fraction of the maximum entropy. This makes the deployment threshold more robust as the number of targets changes, for example.

We performed a series of experiments to test the effectiveness of this entropy-based threshold approach. There are two UAVs, each of which carries 20 UGSs. When a UGS is deployed, it has a life span of 3,000 time periods, after which its sensor dies. There is a set of five targets to be detected. Once a target is detected by a UAV, it is immediately replaced with a new target that has a location chosen at random uniformly, and the UAVs know this initial location. This target search with replacement continues for 10,000 time periods, and then the simulation ends.

The baseline behavior is to have the UAVs deploy UGSs uniformly over time. That is, for 20 UGSs and a 10,000 period simulation, the UGSs are deployed wherever the UAV happens to be at times 0, 500, 1000, 1500, …, and 9500. Note that the final deployment at time 9500 is somewhat of a waste because the UGS will be active only for 500 periods, even though it has a lifetime of 3000 periods. Given this baseline behavior, we track two statistics, the average location error per target over time and the number of targets detected throughout the simulation. We average these statistics over ten independent trials to establish the baseline.

As a comparison against this baseline, we have UAVs that deploy UGSs when the entropy exceeds a fraction H of the maximum entropy. Consider the two extreme cases. If $H = 0$, then each UAV deploys a UGS whenever the entropy exceeds zero, which means that all UGSs are deployed early in the simulation and expire long before the end of the simulation. On the other extreme, if $H = 1$, then a UGS would be deployed only when the entropy is equal to the maximum entropy, which will never happen in practice. This behavior means that the simulation will end with no UGSs being deployed.

We consider values of *H* = 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6 and 0.7. For ten independent trials, we record the average target location error over time and the number of detections for each value of *H*, and then express these two metrics as a percentage of the baseline value. That is, if the average error for the baseline is 50 and the average error for *H* = 0.1 is 45, then the scaled average error is reported as 0.9.

Figure 6–3 shows the results of the experiment. For extreme values of *H*, the average error is higher than the baseline and the number of detections is lower than the baseline. However, for *H* = 0.2, 0.3 and 0.4, the deployment threshold approach beat the baseline. For the best case (*H* = 0.3), the detection rate was 12 percent higher than the baseline and the error was 13 percent lower than the baseline.

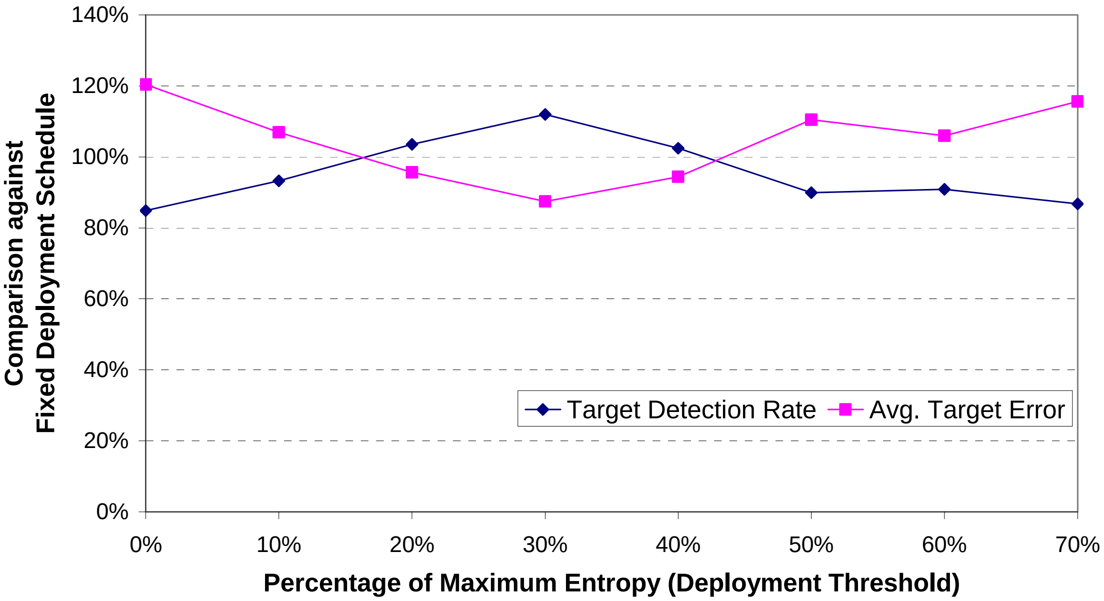


Figure 6–3: Experimental results using a set of fixed deployment thresholds

6.1.2. UGS deployment approach (dynamic threshold)

There are some disadvantages of the fixed deployment threshold approach. In particular, it fails to address the problem of pacing. For example, if the fixed threshold is such that the UAV has several UGSs remaining near the end of the simulation, then the UAV should lower its deployment threshold (assuming for simplicity that there is no value in holding UGSs in inventory at the end of the simulation). Similarly, if the fixed threshold is such that the UAV is deploying UGSs too liberally early in the simulation, then the UAV should raise its threshold in order to slow the deployment rate.

Ideally, we would have a dynamic deployment threshold $H(S, T)$, expressed as a fraction of the maximum entropy, where S is the number of remaining UGSs and T is the amount of time remaining in the simulation. We developed an approach to do this, based on the illustration in Figure 6–4. We assume that the entropy can be modeled reasonably well by a probability distribution with probability distribution function $f(h) = be^{-bh}$ and cumulative distribution function $F(h) = 1 – e^{-bh}$. The shaded area under the curve in Figure 6–4 reflects the probability that the entropy in a given location is larger than H, which may also be thought of as the probability of deploying a UGS given a threshold H.

Our idea for the dynamic threshold is to set the threshold such that the probability of deployment matches the acceptable deployment pace. For example, suppose $H = 0.4$ and the shaded area equals two percent. This implies that there will be roughly two deployments per hundred periods in the simulation. If there are ten UGSs remaining and 1000 periods to go, then a rate of two deployments per hundred periods is too fast, so H will have to increase. If there are forty UGSs remaining and 1000 periods to go, then that rate is too slow and H will have to decrease.

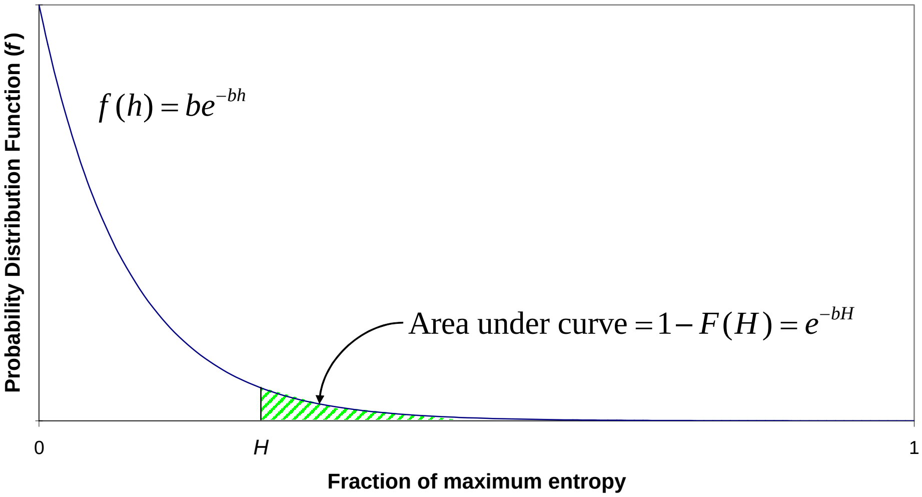


Figure 6–4: Dynamic thresholds based on a probability distribution

Given limited funds available for this extension, we made some simplifying assumptions in order to develop an approach for testing. We start by choosing a suitable value of b at the start of the simulation. We do this by choosing a threshold H that performs well as a fixed threshold, and then use the values S_0 and T_0 at the start of the simulation to solve for b,

$$1 - F(H) = e^{-bH} = \frac{S_0}{T_0} \quad \Rightarrow \quad -bH = \left(\ln S_0 - \ln T_0\right) \quad \Rightarrow \quad b = \frac{\ln T_0 - \ln S_0}{H}. \qquad (6.2)$$

For the experiments in Figure 6–3, H = 0.3, S = 20 and T = 10,000, so the value of b that we would use is b = 20.7. Next, we use this value of b along with the values of $S(t)$ and $T(t)$ at any time t in the simulation to compute the threshold $H(t)$ to use,

$$-bH(t) = \left(\ln S(t) - \ln T(t)\right) \quad \Rightarrow \quad H(t) = \frac{\ln T(t) - \ln S(t)}{b}. \tag{6.3}$$

Figure 6–5 shows the results of using the dynamic deployment threshold with an initial threshold H = 0.3. The fixed deployment threshold curves are the same as in Figure 6–3. The dynamic threshold easily outperforms all of the fixed thresholds, increasing the detection rate by slightly over 20 percent above the baseline and decreasing the average target error by nearly 30 percent.

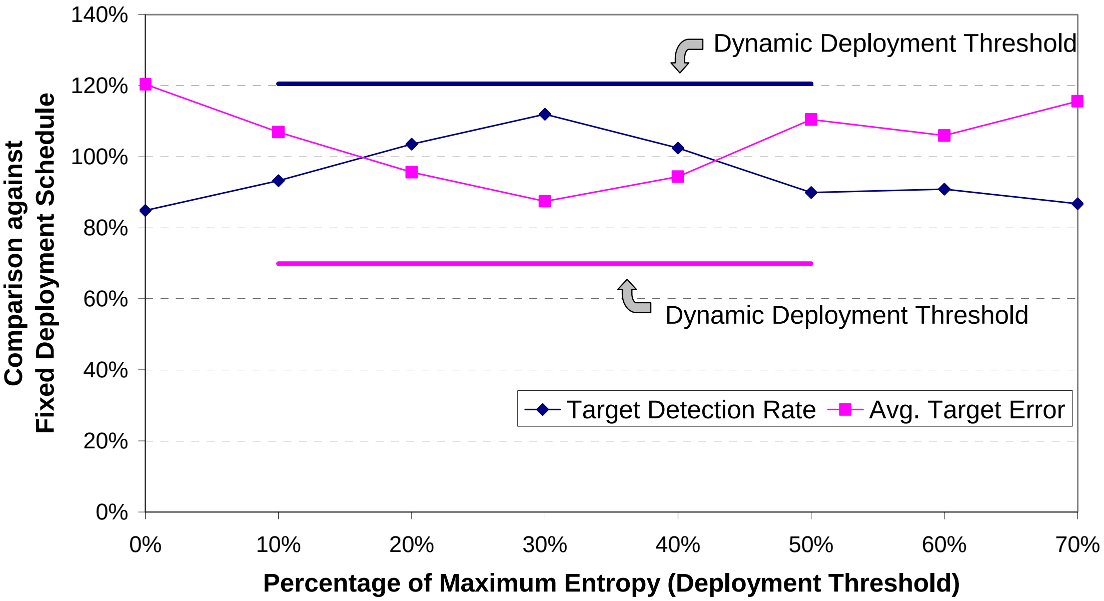


Figure 6–5: Comparison of dynamic deployment thresholds with a set of fixed thresholds

Although the dynamic threshold approach that we developed makes a number of simplifying assumptions (e.g., entropy values follow exponential distribution, need to know which fixed threshold is effective in order to initialize the dynamic threshold), we believe that this approach shows great promise for developing more sophisticated deployment strategies based on computing the entropy.

6.2. EVASIVE TARGET MODELS

In this section, we describe an approach by which the UAVs can improve their effectiveness at detecting evasive targets. We start by describing the evasive motion model.

6.2.1. Definition of evasive motion

The evasive motion model that we derive is a linear combination of evasive motion based on the location of each of the UAVs relative to a given target and random motion as before. The relative weights of the evasive and random components depend on the proximity of the UAVs to the target.

Let the vectors $\mathbf{u}_1,\ldots,\mathbf{u}_M$ denote the locations of the M UAVs, and the vector $\mathbf{v}$ denote the location of a target. The resultant vector $\mathbf{w}$ for the target (see Figure 6–6) is given by

$$\mathbf{w} = \sum_{m=1}^{M} \frac{-(\mathbf{u}_m - \mathbf{v})}{|\,\mathbf{u}_m - \mathbf{v}\,|^3}\,. \tag{6.4}$$

The m^{th} term in this sum is a vector in the direction opposite that of $\mathbf{u}_m$ from $\mathbf{v}$, of magnitude $|\,\mathbf{u}_i - \mathbf{v}|^{-2}$. This weighted sum of “repulsive” vectors for the target at $\mathbf{v}$ leads to a resultant direction in which the target can move to evade the UAVs.

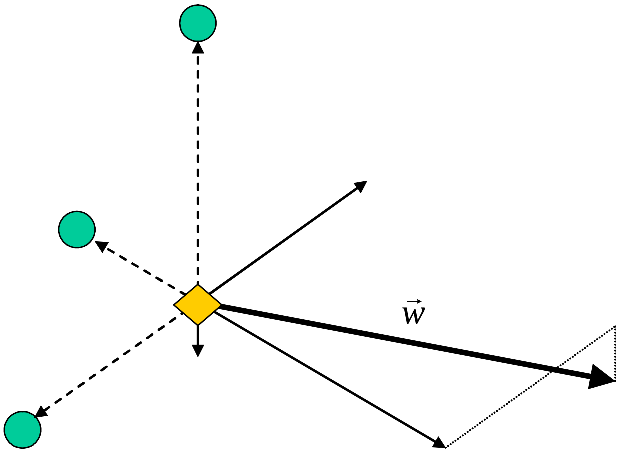


Figure 6–6: Resultant vector for evasive motion by the target

However, since we have a rectangular simulation area, evasion via repulsive vectors will tend to have targets collect in the corners of the search area. Rather than convert our search area into a torus (donut shape) that does not contain any corners in which the target could collect, we instead add repelling forces to keep the target away from the corners. Given the bounding box for the search area (0, 0) and (x_{max}, y_{max}), we write the final form of the resultant vector **w** as follows, where we write **v** = (v_x, v_y),

$$\mathbf{w} = \frac{1}{5}\cdot\left[\frac{\left(v_x,0\right)}{5\left|v_x\right|^3} + \frac{\left(-(x_{\max}-v_x),0\right)}{5\left|x_{\max}-v_x\right|^3} + \frac{\left(0,v_y\right)}{5\left|v_y\right|^3} + \frac{\left(0,-(y_{\max}-v_y)\right)}{5\left|y_{\max}-v_y\right|^3}\right] + \sum_{m=1}^{M}\frac{-(\mathbf{u}_m-\mathbf{v})}{|\,\mathbf{u}_m-\mathbf{v}\,|^3} \quad (6.5)$$

The first four terms of equation (6.5) can be viewed as repulsion vectors deriving from imaginary UAVs placed at (v_x, 0), (v_x, y_{max}), (0, v_y), and (x_{max}, v_y), each with 1/5 of the repulsive force of an actual UAV (the value 1/5 was chosen empirically based on watching the evasive motion on the screen for different weights).

The direction of the resultant vector **w** determines the evasive direction of the target motion, and its magnitude determines the probability p_e of an evasive step in the next time period instead of the random step. We select a parameter λ, typically between 2000 and 20000, and define the probability p_e by

$$p_e \;=\; 1-e^{-\lambda|\mathbf{w}|}. \quad (6.6)$$

The target then has probability p_e of moving one step of fixed length in the direction of the resultant **w**, and probability 1 – p_e of taking one step of fixed length in a random direction.

6.2.2. Evasive motion model for updating the prior distribution

In addition to describing the fixed step motion of the target, we need to derive a motion model to be used to update the prior distribution on target location. The motion model that we use is analogous to that derived in Appendix E.2, which defines a probability q that a target transitions out of its current cell in the next time step.

Figure 6–7 shows the evasive transition function that we will use to model the evasive motion. There are two components, evasive and random. Both components assume a probability q of transitioning out of the current cell in the next time step. The evasive component puts all of the weight q in the cell closest to the direction of **w**, and the random component spreads the weight q evenly across the six neighbors. We then use the probability p_e to take a weighted average of the evasive and random transition functions.

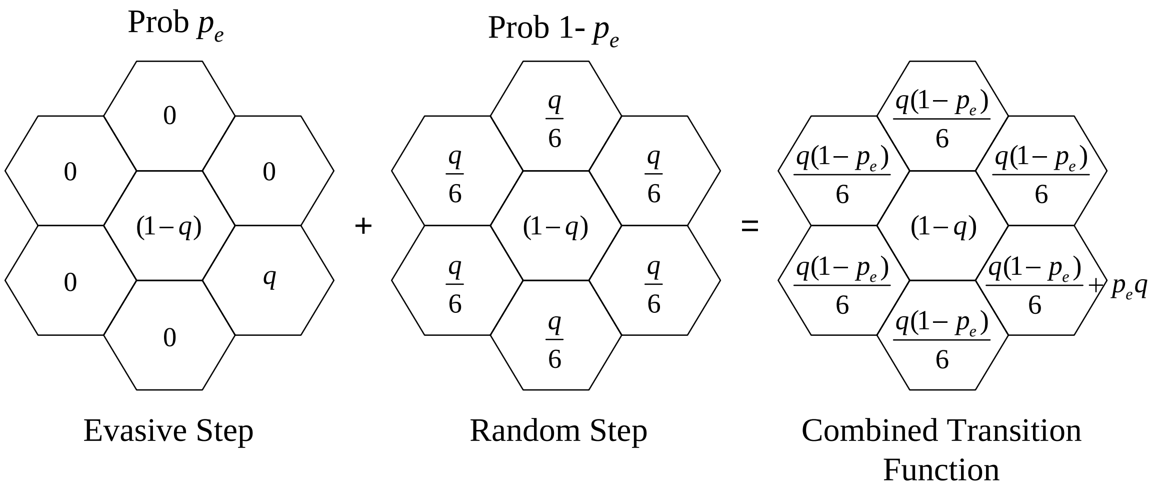


Figure 6–7: Derive transition function as a weighted average of evasive and random motion

6.2.3. Experimental results

We performed a set of experiments to evaluate the search performance when targets moved randomly or evasively and when UAVs assumed the target motion was either random or evasive. We considered all four combinations of number of UAVs (either two or four) and number of targets (either five or ten). Once a target is detected, it is replaced immediately with a new target that has a location chosen at random uniformly, and the UAVs know this initial location. Using limited trial and error, we chose a value of λ = 5000. Each simulation consisted of 10,000 periods, and statistics on the number of target detections were collected over periods 5,000-10,000 to allow the simulation to stabilize over the first 5,000 periods.

Figure 6–8 shows the results over ten independent trials. The target follows either purely random motion or the evasive motion model derived in this section. The UAVs update the target prior distributions based on assuming either purely random motion or the evasive motion model. Clearly, the detection rate is the highest when the targets move randomly and the UAVs model this random motion. If the UAVs incorrectly model this random target motion as evasive, then the detection rate drops to 22 to 42 percent of the original rate.

If the targets follow the evasive motion model and the UAVs correctly model this evasive motion, then the detection rate is similar to the case where targets move randomly and UAVs model evasive motion. If the UAVs incorrectly model the evasive motion as random, then the detection rate drops even more dramatically than before to 10 to 26 percent of the original rate.

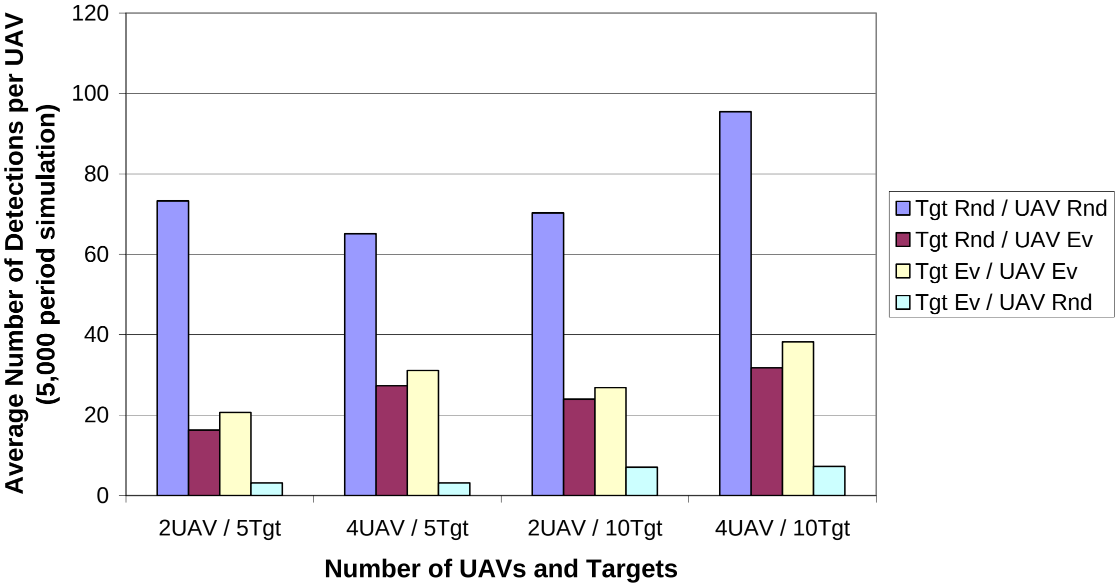


Figure 6–8: Experimental results comparing evasive and random motion models

From a game theoretic standpoint, it is interesting to note the choice of target motion model that a UAV must make. If the UAVs choose to model the target motion as random, then the UAVs perform extremely well if right and extremely bad if wrong. On the other hand, if the UAVs choose to model the target motion as evasive, then the detection rate is nearly the same whether the target moves evasively or randomly.

One area of future research that we would like to pursue is how to improve UAV coordination to make the search for evasive targets more effective. In particular, we would like to investigate an approach by which the UAVs focus on containing an evasive target rather than strictly maximizing the probability of detection. Given a Gaussian prior distribution, all UAVs fly first to the center of the distribution where the probabilities are the highest. However, an evasive target that eludes initial detection can escape easily because the search paths leave large corridors to exploit.

Instead, the UAVs could collaborate on a joint plan to surround and contain the prior and flying an ever-decreasing spiral to tighten the distribution over time. This leads to a lower probability of detection in the initial effort, but guarantees a higher probability of detection over the longer term. The keys to effective containment are: (1) identifying containment opportunities, (2) switching the local goals of the UAVs from detection to containment, and (3) reverting to detection again once the containment reaches a critical threshold.

6.3. JOINT COORDINATED UAV SEARCH AND SURVEILLANCE

The final extension puts the search and surveillance roles together for another layer of coordination among the UAVs. The idea is that at a given time, a UAV can only be in search (detection) or surveillance (monitoring) mode. However, the UAV can switch between the modes over time. When a search UAV detects a target, responsibility for the target is passed to a surveillance UAV to maintain localization. When a surveillance UAV loses a target, responsibility for the target (along with information about where and when the target was last located) is passed to the search UAVs for re-detection.

The interesting research question is whether the autonomous fleet of UAVs can self-organize dynamically and reach the optimal mix of search and surveillance effort given various environmental stresses (more targets or fewer targets, faster targets or slower targets, more UAVs or fewer UAVs, bigger sensor footprints or smaller sensor footprints). Our approach is based on having each UAV estimate the marginal value of continuing its current role versus switching to the other role.

6.3.1. Asymptotic analysis of search and surveillance roles

Under the approach we developed, UAVs switch roles autonomously (without outside direction) between search and surveillance based on the marginal value of performing each role. Each surveillance UAV estimates the expected increase in location error for the surveillance targets and the expected decrease in location error for the search targets if that UAV switched from a surveillance role to a search role. The basic rule states that a UAV switches if the net expected error decreases. In this section, we derive the underlying functions used to apply this rule.

We assume that the UAVs know four types of information at a particular time. For surveillance, the UAVs will know: (1) the number of surveillance UAVs, M_{Surv}, (2) the number of surveillance targets, N_{Surv}, (3) the surveillance target speed v, and (4) the cumulative amount of time since each of the surveillance targets was last detected, T_{Surv}. This last statistic merits further explanation. Each surveillance target n has gone some time t_n since it was last detected. The cumulative time since last detection is the sum of these times, $T_{Surv} = \sum_{n=1}^{N_{Surv}} t_n$. For search, the UAVs know similar statistics, M_{Srch}, N_{Srch}, v and T_{Srch}. Note that, in this case, we assume that the target speed v stays the same in each mode.

Using a methodology similar to that in Appendix C, we can derive an estimated root-mean-squared (RMS) target error for the set of targets in either mode. Since this estimate is true for either mode, we drop the "*Surv*" and "*Srch*" subscripts for now. For the Pearson random walk model with step size *v*, the expected squared distance from the initial target position after *t* time periods is $v^2 t$. Given *N* targets and a cumulative time since last detection *T* for those targets, the average time since last detection is *T*/*N*.

Let us assume that the time since last detection for target *n* is $t_n \sim U\left[0, 2T/N\right]$; that is, the time varies uniformly between zero and double the mean. For targets in a surveillance tour, this distribution is reasonable, as is shown in Appendix D. Given this distribution, we can compute the expected RMS distance from the initial target position to be

$$E\left[d^{RMS}\right] = \sqrt{\frac{1}{2T/N}\int_0^{2T/N} v^2 t\, dt} = \sqrt{\frac{Nv^2}{2T}\left[\frac{t^2}{2}\right]_0^{2T/N}} = v\sqrt{\frac{N}{2T}\cdot\frac{1}{2}\frac{4T^2}{N^2}} = v\sqrt{T/N}\ .$$

The expected sum, *D*, of the RMS distance errors across all *N* targets is given by

$$D \;=\; E\left[\sum_{n=1}^{N} d_n^{RMS}\right] \;=\; Nv\sqrt{T/N} \;=\; v\sqrt{N^2T/N} \;=\; v\sqrt{NT}\ . \qquad (6.7)$$

Equation (6.7) can be used to compute D_{Surv} and D_{Srch} for the two modes using the respective values of *v*, *N* and *T*. The objective of the UAV fleet, then, is to minimize the sum of the target location errors for each mode,

$$\min_{M_{Surv}, M_{Srch}} D_{Surv} + D_{Srch},\ \text{subject to } M_{Surv} + M_{Srch} = M\ . \qquad (6.8)$$

However, the problem is that D_{Surv} and D_{Srch} do not depend explicitly on M_{Surv} and M_{Srch}, respectively. Instead, we rely on empirical observations.

During initial empirical testing and debugging, we discovered an interesting property. Given a mix of roles such that $M_{Surv} + M_{Srch} = M$, we let the system evolve until the location errors in both modes settled to an asymptotic value. We observed that $M_{Surv}{\cdot}D_{Surv}$ and $M_{Srch}{\cdot}D_{Srch}$ were relatively stable for different values of M_{Surv} and M_{Srch}, subject to the sum equaling *M*. Let K_{Surv} and K_{Srch} denote these products, which in full form are written as

$$K_{Surv} = M_{Surv}D_{Surv} = vM_{Surv}\sqrt{N_{Surv}T_{Surv}} \quad \text{and} \quad K_{Srch} = M_{Srch}D_{Srch} = vM_{Srch}\sqrt{N_{Srch}T_{Srch}}\ .$$

We can recharacterize the optimization problem in equation (6.8) as

$$\min_{M_{Surv}, M_{Srch}} \frac{K_{Surv}}{M_{Surv}} + \frac{K_{Srch}}{M_{Srch}}, \text{ subject to } M_{Surv} + M_{Srch} = M\ . \tag{6.9}$$

Under this new form, we compute the values of K_{Surv} and K_{Srch} using the current values of M_{Surv} and M_{Srch}, and then solve the optimization problem by fixing the K values and allowing the M values to vary.

This optimization problem can be solved directly by rearranging the constraint as $M_{Srch} = M - M_{Surv}$ and substituting. We can rewrite the optimization problem as

$$\min_{M_{Surv}} g\left(M_{Surv}\right), \text{ where } g\left(M_{Surv}\right) = \frac{K_{Surv}}{M_{Surv}} + \frac{K_{Srch}}{M - M_{Surv}}\ . \tag{6.10}$$

After rearranging the terms, we take the derivative of g,

$$\begin{aligned} g\left(M_{Surv}\right) &= K_{Surv}\left(M_{Surv}\right)^{-1} + K_{Srch}\left(M - M_{Surv}\right)^{-1} \\ g'\left(M_{Surv}\right) &= K_{Surv}\left(-1\right)\left(M_{Surv}\right)^{-2} + K_{Srch}\left(-1\right)\left(M - M_{Surv}\right)^{-2}\left(-1\right) \\ &= \frac{-K_{Surv}}{M_{Surv}^2} + \frac{K_{Srch}}{\left(M - M_{Surv}\right)^2}. \end{aligned}$$

Setting the derivative to zero and solving for M_{Surv}^*, we get

$$\begin{aligned} \frac{K_{Surv}}{\left(M_{Surv}^*\right)^2} &= \frac{K_{Srch}}{\left(M - M_{Surv}^*\right)^2} \\ \left(\frac{M - M_{Surv}^*}{M_{Surv}^*}\right)^2 &= \frac{K_{Srch}}{K_{Surv}} \\ \frac{M}{M_{Surv}^*} - 1 &= \sqrt{\frac{K_{Srch}}{K_{Surv}}} \\ \frac{M}{M_{Surv}^*} &= 1 + \sqrt{\frac{K_{Srch}}{K_{Surv}}} \\ M_{Surv}^* &= \frac{M}{1 + \sqrt{\frac{K_{Srch}}{K_{Surv}}}}\ . \end{aligned} \tag{6.11}$$

Although equation (6.11) allocates search and surveillance effort optimally, there is no guarantee that M^*_{Surv} is an integer. Consequently, we propose a more practical optimization procedure. At time *t*, one UAV is given an opportunity to change its role. If the UAV is performing surveillance, then it will switch to search if

$$\frac{K_{Surv}}{M_{Surv}-1}+\frac{K_{Srch}}{M_{Srch}+1} \quad < \quad \frac{K_{Surv}}{M_{Surv}}+\frac{K_{Srch}}{M_{Srch}}. \tag{6.12}$$

If the UAV is performing search, then it will switch to surveillance if

$$\frac{K_{Surv}}{M_{Surv}+1}+\frac{K_{Srch}}{M_{Srch}-1} \quad < \quad \frac{K_{Surv}}{M_{Surv}}+\frac{K_{Srch}}{M_{Srch}}. \tag{6.13}$$

Note that in either case, the number of surveillance or search UAVs must never become zero.

6.3.2. Experimental results for asymptotic analysis

We perform a series of experiments to measure the effectiveness of the switching criteria. Start with 50 search targets and zero surveillance targets. We track the average target location error over time for three scenarios: (1) a fixed mix of one search UAV and nine surveillance UAVs, which we call 1:9, (2) a fixed mix of four search UAVs and six surveillance UAVs, which we call 4:6, and (3) a dynamic mix based on the switching criteria in equations (6.12) and (6.13). Figure 6–9 shows the average target location error for each mix on the primary axis and the number of surveillance UAVs over time for the dynamic mix on the secondary axis.

The error for the fixed 1:9 mix grows quickly (with a peak of about 13 units) over the first 500 periods because there is only one search UAV to perform the initial target detections. The error then drops over the next 1,000 periods as the surveillance UAVs maintain tight estimates of the target locations with few losses to be handled by the sole search UAV. The asymptotic error (around six units) is extremely low because the optimal mix of effort in the limit is to maximize the number of surveillance UAVs.

For the fixed 4:6 mix, the initial detections are made rather quickly (reducing the peak error to nine units) because there is sufficient search effort. However, six surveillance UAVs are inadequate for maintaining the surveillance target locations, so the asymptotic error only settles down to around seven units due to the higher loss rate on surveillance targets.

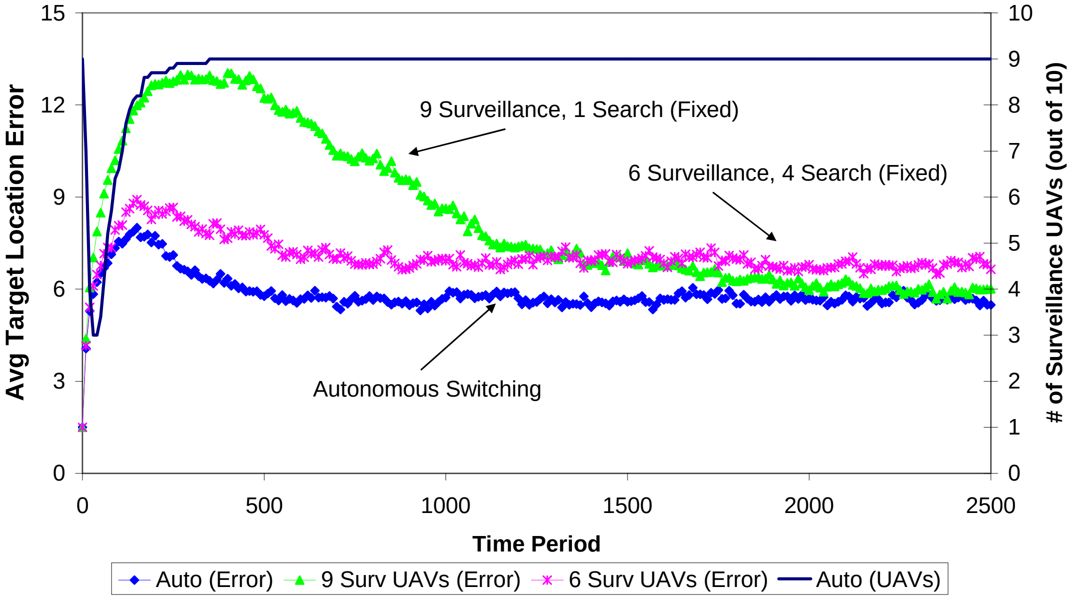


Figure 6–9: Comparison of fixed roles versus autonomous switching over 2,500 periods

The autonomous switching starts with nine surveillance UAVs and one search UAV. Over the first fifty periods, the search effort increases automatically as surveillance UAVs switch to search mode, reaching a maximum of seven search UAVs. The peak target location error is only eight units because the system has adjusted quickly to the need for additional search effort. Once most of the search targets have been detected, the system adjusts automatically by switching gradually all but one of the search UAVs back to surveillance with the asymptotic error falling to six units. This robust behavior illustrates that the UAV fleet can self-adjust and self-organize their roles in response to the environment.

To investigate this behavior further, we applied a wide range of environmental parameters. For a particular set of parameters, we compared the limiting behavior (the average target location error over the last 5,000 periods of a 10,000 period simulation) using the autonomous switching algorithm against a range of fixed mixes of search and surveillance UAVs that do not change over the course of the simulation. Figure 6–10 shows the asymptotic results from these trials.

For example, given the baseline simulation of ten UAVs and 100 targets, a fixed mix of two search UAVs and eight surveillance UAVs yields an average error of 28.5 units. For the fixed allocations, a roughly equal mix (4:6, 5:5, 6:4) has a low target error. The autonomous switching runs had an error of 22.0 units for this baseline case, which is close to the best of the fixed mix runs without the need to predict in advance which mix is optimal.

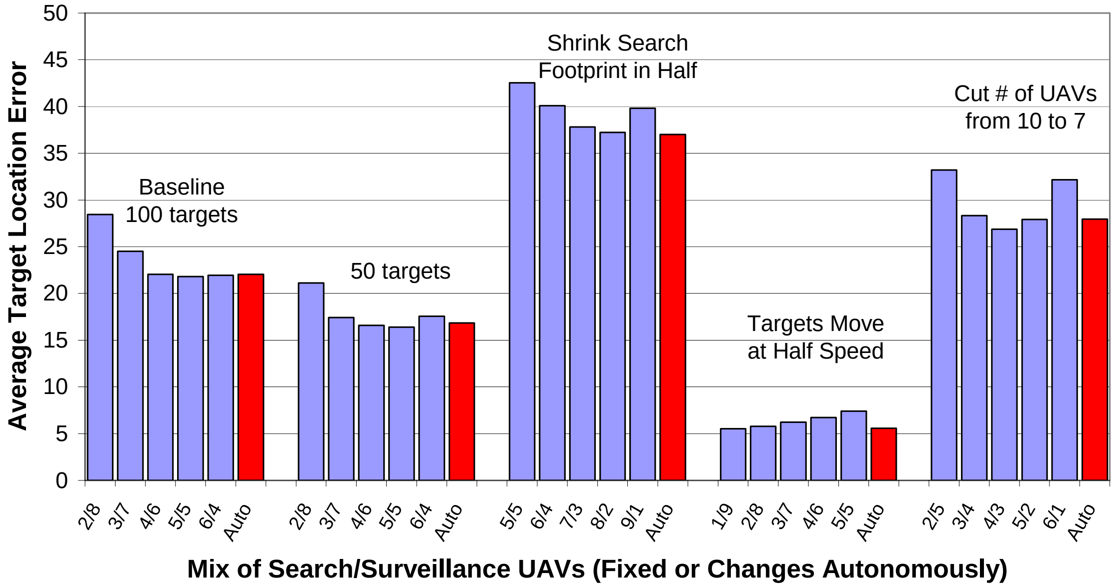


Figure 6–10: Asymptotic comparison of fixed roles versus autonomous switching given different environmental parameters

Looking at the five environmental scenarios in Figure 6–10, we see that the optimal mixes are 5:5, 5:5, 8:2, 1:9 and 4:3, respectively. In particular, there is no fixed mix that performs well in all environments. However, the autonomous switching is extremely robust, with an error comparable to the best of the fixed mix results for all cases.

6.3.3. Ideas for real-time analysis of role switching

Although the autonomous switching algorithm performed well in the asymptotic experiments, there are opportunities for improvement. For example, this switching approach can lead to instability if all UAVs are given an opportunity to switch every period because if one search UAV computes that there is an incentive to switch to surveillance, then all of the search UAVs will also decide to switch to surveillance. Allowing only one UAV per time period to be eligible to switch roles reduces this instability.

Another potential problem is oscillatory behavior in which the system flips repeatedly between two states. For example, if the optimal allocation is 3.5 surveillance UAVs, then the system may toggle between three and four surveillance UAVs. Using the switching rules defined in equations (6.12) and (6.13), it is not unusual to have a surveillance UAV switch to search and then, a few periods later, switch back to surveillance when the marginal values are nearly the same.

One approach to remedy this oscillation is to introduce some friction (or transaction cost) into the decision to switch roles. Switching roles can add short-term disruption to the system. For example, losing a surveillance UAV is disruptive because all of its targets must be reallocated to the other surveillance UAVs. Adding a surveillance UAV is also disruptive because it takes some time to reallocate the targets and balance the workload across UAVs.

We have modified the switching rules in equations (6.12) and (6.13) to include a switching threshold, ε, that requires a greater reduction in location error than simply exceeding the point of indifference. If the UAV is performing surveillance, then it will switch to search if

$$\frac{\dfrac{K_{Surv}}{M_{Surv}-1}+\dfrac{K_{Srch}}{M_{Srch}+1}}{\dfrac{K_{Surv}}{M_{Surv}}+\dfrac{K_{Srch}}{M_{Srch}}} \; < \; 1-\varepsilon\,. \tag{6.14}$$

If the UAV is performing search, then it will switch to surveillance if

$$\frac{\dfrac{K_{Surv}}{M_{Surv}+1}+\dfrac{K_{Srch}}{M_{Srch}-1}}{\dfrac{K_{Surv}}{M_{Surv}}+\dfrac{K_{Srch}}{M_{Srch}}} \; < \; 1-\varepsilon\,. \tag{6.15}$$

For example, if $\varepsilon = 0.05$, then a surveillance UAV would switch to search only if it would decrease the expected target location error by at least five percent. If ε is set too high, then opportunistic switches do not occur. If ε is set too low, then oscillatory switching occurs too frequently. What we have found is that setting the threshold in the five to ten percent range ($\varepsilon = 0.05$ to 0.10) provides asymptotic results comparable to the runs presented earlier, with more robust changes in the UAV ratios when environmental parameters change.

In the next chapter, we discuss two transitions of the TASK technology that we have developed and present our conclusions on the research.

7. TECHNOLOGY TRANSITIONS AND CONCLUSIONS

Metron has two official transitions of the UAV search technology, one to a DARPA DSO seedling contract and the other to Naval Air Systems Command (NAVAIR) Phase I and II SBIR contracts. In this chapter, we discuss both transitions briefly, and then provide closing thoughts on our TASK research.

7.1. DARPA DSO TRANSITION

Metron was awarded a seedling contract, entitled “Top-Down Mechanism Design Study for Multi-UAV Search and Surveillance” (contract W911NF-04-C-0041), by DARPA DSO to investigate “function-driven design” technology for complex distributed systems, and to apply these technologies to a UAV ground target surveillance scenario. The research effort was supervised by Dr. Carey Schwartz, DARPA DSO program manager.

There were two primary breakthroughs in our research. The first is a value potential approach to optimizing search paths based on approximating an infinite-horizon search plan. Using this value potential to dictate UAV motion improves the search performance, especially for disjoint, multimodal (“patchy”) probability distributions on target position.

The second innovation introduces dynamic area sectoring, which allows UAVs to partition the search area dynamically and to balance the search workload across UAVs. Sectoring also eliminates the need to deconflict search paths and simplifies collision avoidance because each UAV stays inside its sector. We summarize each innovation below, and the full details are available as part of the final technical report [God05].

7.1.1. Value potential for search path planning

In section 5.5, we identified several shortcomings of finite-horizon planning, including the inability to find gradients in low probability areas. As part of the DSO study, we investigated ways to estimate the infinite-horizon potential search value associated with having a UAV in cell *i* at time *t*. This estimate can provide a basis for choosing to move to the cell to the left, right or straight-ahead, even if the neighboring cell probabilities are low. We will label the set of feasible moves from cell *i* as $\{i_L, i_R, i_S\}$, respectively.

We considered two forms of these value potential functions that adjust the probability of detecting a target in a cell by the distance of that cell from the current UAV location. The first form, called the "Lambda" form, is similar to the discounting that was described previously. The Lambda form uses the following rule,

$$\max_{i' \in \{i_L, i_R, i_S\}} p^D \cdot \sum_{j \in I} \lambda^{d_{i'j}} p_j(t+1) \qquad \text{(Lambda rule)}. \tag{7.1}$$

The Lambda form loops over all cells (the set *I*) and discounts the probability of detecting a target in cell *j* by a factor of $\lambda^{d_{i'j}}$, where $d_{i'j}$ is the distance from cell *i′* to cell *j*, expressed in units of the inter-cell spacing. That is, if cells *i′* and *j* are neighbors, then $d_{i'j} = 1$, and if the cells are not neighbors, then scale by the distance between the centers of two neighbors. For hexagonal cells with side length *s*, neighbors have centers that are $\sqrt{3}\,s$ units apart.

The second form is called the "1/d" form, which uses a different form for the discounting. The "1/d" form uses the following rule,

$$\max_{i' \in \{i_L, i_R, i_S\}} p^D \cdot \sum_{j \in I} \left(\frac{1}{1 + d_{i'j}} \right)^Q \cdot p_j(t+1) \qquad \text{(1/d rule)}. \tag{7.2}$$

Figure 7–1 shows the influence of these value potential surfaces on the search planning. The picture on the far left is the original prior p_i at some time. The top row of pictures shows the value potential surface associated with the "1/d" form defined in equation (7.2) for different powers *Q*. The value potential is computed for every cell *i* in the search area (as if the UAV were located in cell *i*), and the color shades are rescaled to these potential values. The bottom row of pictures shows the value potential surface associated with the Lambda form defined in equation (7.1) for different discount factors *λ*.

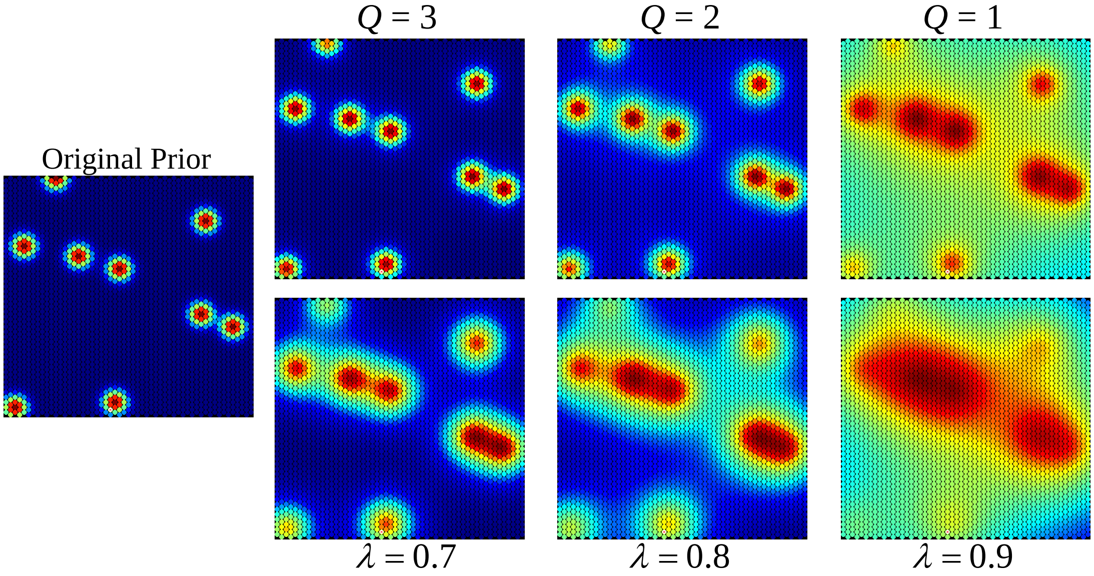


Figure 7–1: Value potential surfaces for different functional forms and parameters

For large values of λ or small values of Q, the prior is smoothed out significantly, and a noticeable gradient is available everywhere. However, look at the mode in the lower left hand corner for the $\lambda = 0.9$ case. The mode has little influence on the value potential surface. If there was a UAV in this vicinity, then it would have to be very close to the center of the mode in order to continue searching that area. In addition, once a UAV is within a high-probability area, the UAV can be drawn away easily due to the influence of other nearby gradients, thereby giving up the search of that area prematurely.

We performed a series of experiments to compare the search performance of three UAVs searching for a single stationary target using the value potential functions. We compared both functional forms: λ^d for $\lambda = \{0.6, 0.7, 0.8, 0.9\}$ and $(1/d)^Q$ for $Q = \{0.5, 1, 2, 3\}$. For comparison, we also included pure 1-step and 5-step finite-horizon path planners, which were used in the TASK research. The UAVs share all sensor data and deconflict paths.

For the environmental setup, there are $k = 9$ modes in the initial prior distribution with a standard deviation of $\sigma = 45$ units split across the nine modes (i.e., each mode has $\sigma_k = 15$ units). The nine modes are distributed at random uniformly throughout the search area. The initial locations of the three UAVs are also randomly drawn from a uniform distribution.

For each of the search path planning approaches, we record the target detection time for 3,200 independent trials. Figure 7–2 shows the median time to target detection, along with the 25th and 75th percentiles, for these approaches.

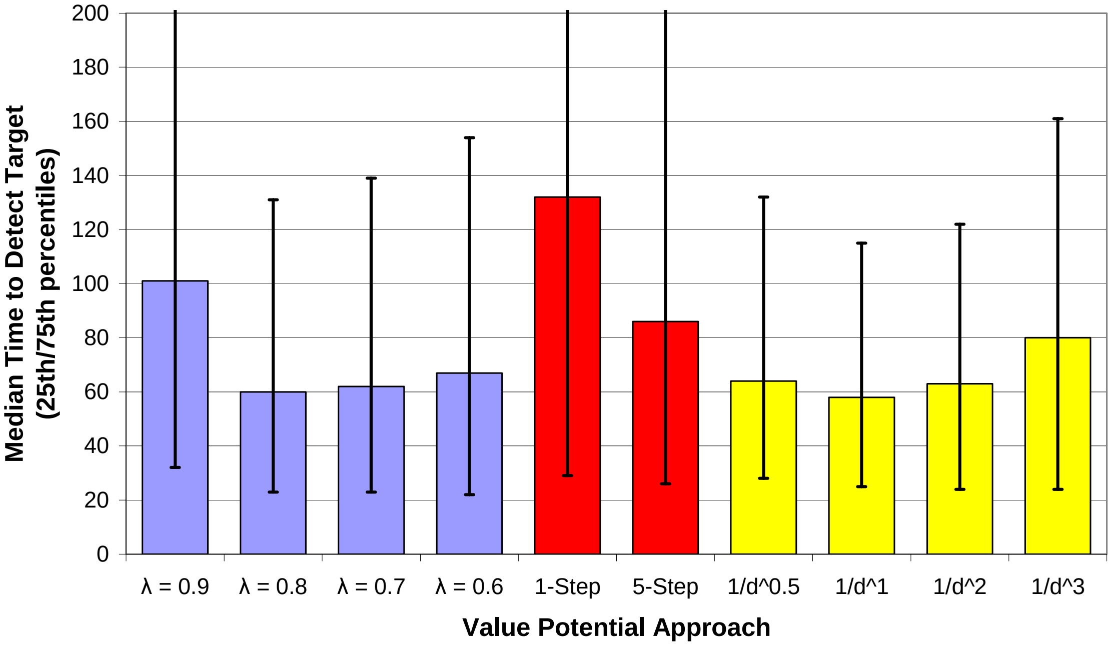


Figure 7–2: Median time to target detection for different value potential approaches

In general, the value potential planners outperformed the finite-horizon planners, detecting targets faster over a wide range of λ and Q parameters. This robust performance is encouraging because it does not require ad hoc testing and simulation to determine a suitable parameter value for a given environmental setting.

7.1.2. Bidding mechanism for dynamic sectoring

We also investigated a bidding mechanism based on UAVs negotiating a dynamic partition of the search area, with each UAV dedicated to one sector. This is analogous to a "zone-defense" approach to search. This bidding mechanism is computationally efficient and effective in practice. The idea is to assign each cell to the "closest" UAV, where closeness depends on a distance adjustment factor that each UAV bids to the other UAVs. Overworked UAVs can make cells seem further away by increasing its distance adjustment factor, which causes the sector for that UAV to shrink. Underworked UAVs can achieve the opposite effect by decreasing its distance adjustment factor, which causes that sector to grow.

The goal of the negotiation is to produce sectors that are balanced and compact. By balanced sectors, we mean sectors that have similar workloads, which we define below. Let $m = 1, 2, \ldots, M$ be the labels for each of the UAVs. For UAV m, we can define the workload $w_m(t)$ for sector m at time t to be

$$w_m(t) = \sum_{i \in I_m} \left(p^D \cdot p_i(t) \right) d_{ii_m} \ , \tag{7.3}$$

where I_m is the set of cells in sector *m*, i_m is the reference cell used to compute distances in sector *m*, and d_{iim} is the distance from cell *i* to the reference cell i_m. The reference cell i_m is either the location of UAV *m* or the cell containing the weighted centroid of sector *m*.

One of the conditions of compactness is that sectors must be simply connected, also known as contiguous. That is, a sector is contiguous if any two cells in that sector can be connected by a path of cells also contained in that sector. For example, two disjoint sets of cells are not contiguous because any path of cells connecting the two subsets must contain cells that are not part of that sector.

As an initial partition, each cell is assigned to the closest UAV. This is equivalent to a Voronoi diagram [BKOS00] that takes a set of *M* points and partitions the area into *M* convex polygons. Each polygon contains one of the reference points, and each point inside the polygon is closer to that reference point than any other reference point.

Figure 7–3 shows Voronoi diagrams for two different sets of six UAVs, one with a uniform prior distribution on target location and the other with a non-uniform prior. Each sector has a dominant color (i.e., orange, blue, green, etc.), and each cell in a sector appears as a shade of this base color. Darker shades correspond to lower probabilities and lighter shades correspond to higher probabilities.

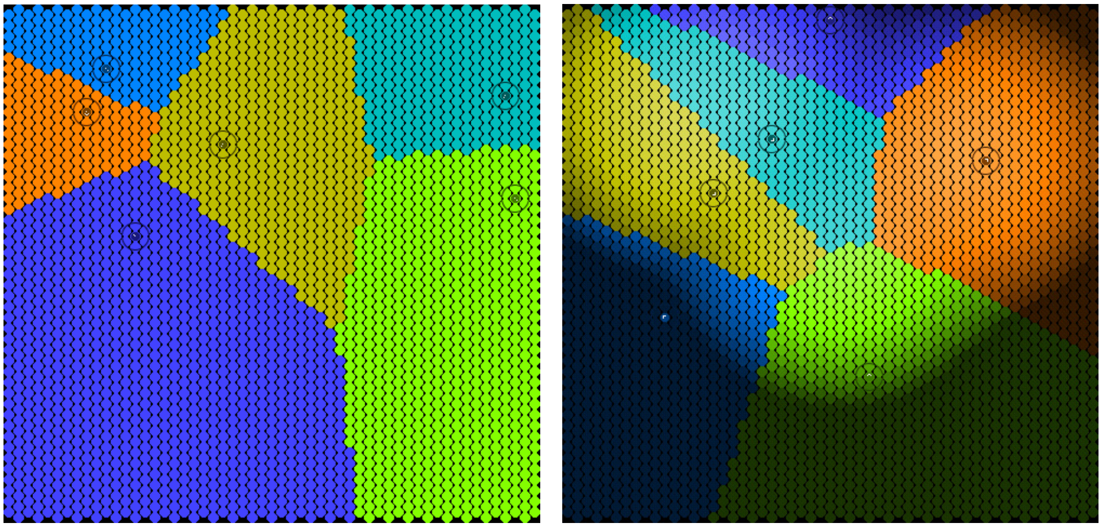


Figure 7–3: Voronoi diagrams for uniform and non-uniform prior distributions

Typically, these partitions lead to sectors of unequal size (workload). For the non-uniform case, the sector in the bottom-left corner has a reasonably large area, but contains little probability. Consequently, that sector has a small workload. As the UAVs move, the Voronoi diagram changes, but the fundamental workload imbalances remain.

We modified the Voronoi criterion so that the UAVs can balance workloads by adjusting boundaries without changing the underlying mathematics. We introduced a distance adjustment factor e^{δ_m} for each UAV, where δ_m is a finite value that may be positive or negative. Let I be a set of cells covering the search area, k, $m \in \{1, \ldots, M\}$ be indices for the M UAVs, and $i_1, i_2, \ldots, i_M$ be reference cells (UAV locations) for each of the M UAVs. The adjusted Voronoi polygon for sector m (the set of cells I_m^δ in sector m) is defined by

$$I_m^\delta = \left\{ i \in I : d_{ii_m} e^{\delta_m} \leq d_{ii_k} e^{\delta_k} \text{ for all } k \neq m \right\}. \tag{7.4}$$

where the distances are measured from the center of one cell to the center of the other cell. Initially, $\delta_m = 0$ (or $e^{\delta_m} = 1$) for all m, which reduces to the original Voronoi form.

At each time step, one UAV is given the opportunity to bid a new δ_m value. We derived two functional forms by which a UAV can determine what value to propose. We show the linear form here. Let $\delta_m(t)$ and $w_m(t)$ be the distance adjustment and workload for UAV m at time t, respectively. Let $w^*(t)$ be the goal workload that the UAV wants to reach, such as the average workload at time t across all UAVs, $w^*(t) = \frac{1}{M}\sum_{m=1}^{M} w_m(t)$. Given this goal, the following equation that can be used to bid the new value $\delta_m(t+1)$,

$$\delta_m(t+1) = \delta_m(t) + K \cdot \left(\frac{w_m(t) - w^*(t)}{W} \right), \tag{7.5}$$

where K is a correction multiplier, and W is an estimate of the asymptotic average workload for each UAV that was derived in the DSO report.

Using the linear updating rule in equation (7.5), we investigated the asymptotic sector boundaries assuming a uniform target prior and no UAV motion. Figure 7–4 shows the sectors that develop given 4, 5, 7, 8, 9, 11, 16 and 25 UAVs. In general, the sectors settle rather quickly, but the time to settle increases as the number of UAVs increases (in part because only a single UAV bids a new δ_m value in each time step).

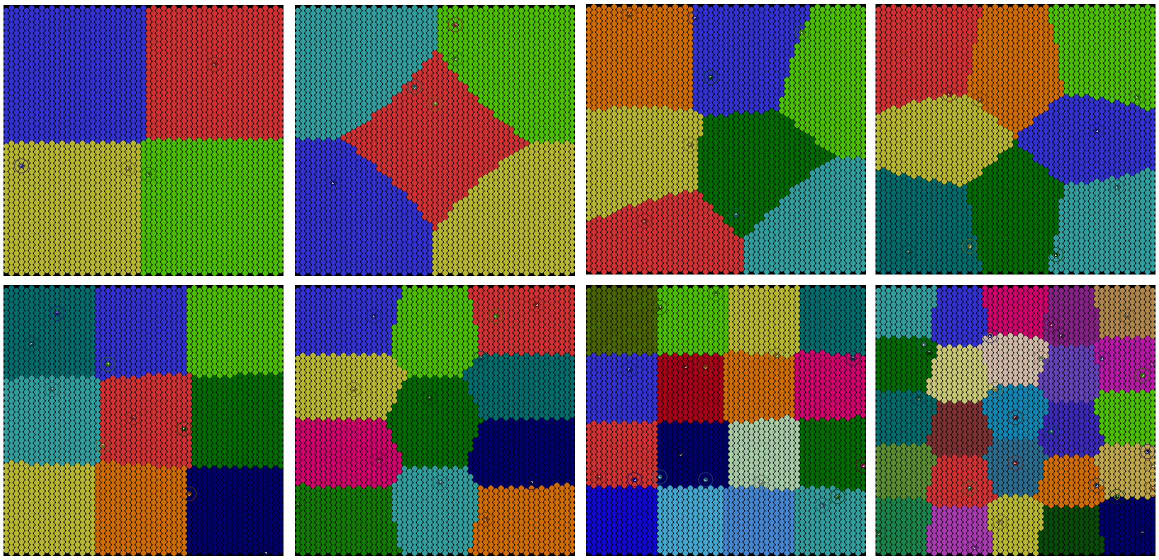


Figure 7–4: Asymptotic sectors given a uniform target prior and no UAV motion

We performed a series of experiments to test the effectiveness of sectoring for both the finite-horizon planner and the value potential planner. Consider three UAVs searching either for one, two or four mobile targets. The initial prior distribution contains sixteen modes across all targets, meaning $k = 16$, 8 or 4 modes per target for one, two or four targets, respectively. The standard deviation of the prior $\sigma = 45$ units is split across the sixteen modes. The modes are distributed at random uniformly throughout the search area, as are the initial locations of the three UAVs. When a target is detected, all modes associated with that target are removed from the probability map.

We record the time to detect all targets for 1,300 independent trials. Figure 7–5 shows the median time to detect all targets, along with the 25th and 75th percentiles, for the different planning approaches, both with and without dynamic sectoring. The finite-horizon search performance is relatively poor, especially as the number of targets increases, because the UAV has difficulty finding gradients that allow the UAV to move from one mode to another. Dynamic sectoring improves significantly the finite-horizon performance because it keeps the UAVs divided across modes. Value potential without sectoring performs well, and dynamic sectoring provides additional improvement.

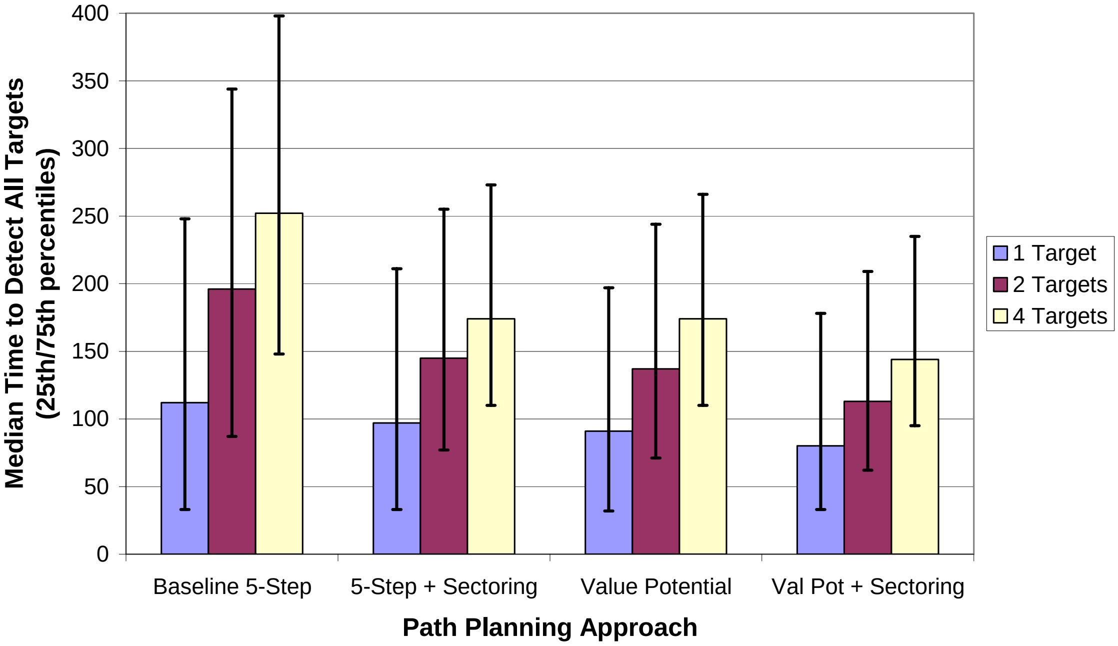


Figure 7–5: Median time to detect multiple moving targets with 16 modes

7.2. NAVAIR SBIR PHASE II TECHNOLOGY TRANSITION

Under SBIR Topic N04-022, entitled "Airborne and Air-Deployed Multi-Sensor Search Optimization," Metron was awarded Phase I and Phase II SBIR contracts (contracts N00014-04-C-0056 and N68335-05-C-0123, respectively) by the Naval Air Systems Command (NAVAIR) to prototype and develop a real-time, air mission planning component into the Undersea Warfare-Decision Support System (USW-DSS) program. This work is being led by Mr. K. C. Stangl, Director of the Charles L. Bartberger M&S Laboratory, Patuxent River.

The main research and development efforts involve combining two Metron core technologies: (1) multi-sensor data fusion based on Likelihood Ratio Tracking (LRT) and (2) coordinated, real-time aircraft search based on distributed optimization. The aircraft search optimization component draws heavily on the research performed under the DARPA TASK and DSO efforts.

The overall objective of the Phase II research and development effort is to transition an air mission planning component into the USW-DSS program. There are three main technical objectives for the Phase II research and development efforts: (1) extend the Phase I integration of the LRT and Air search optimization components, (2) optimize multiple sensor systems for pre-mission planning, and (3) provide initial transition and integration of the developed code into Build 1 of the USW-DSS program.

Under the Phase I research, we combined LRT-based data fusion with coordinated air search optimization, and performed a series of controlled experiments to show the value of integrated real-time mission planning. The full details are available as part of the Phase I Final Report [God04], which we summarize below.

To demonstrate the value of combining LRT with air search optimization, we designed and performed a series of experiments. We assume a square search area with side length 60 nautical miles, partitioned into a hexagonal grid. Each hexagonal cell has area approximately one square nmi. A single mobile target is present, and each simulation run ends with a "detection", defined as when one of the P-3 aircraft is in the same cell as the target.

We simulated data from two types of sensors, a field of sonobuoys and radar from P-3 aircraft. The sensor characteristics were selected to demonstrate the technology and to perform controlled experiments, rather than for realism. In particular, the characteristics were chosen to balance the contributions to the likelihood surfaces made by each sensor type, rather than having the data from one sensor type dominate the data from another type.

Sonobuoy field model. We assume a stationary 4x4 sonobuoy field that processes data every two minutes. At each time update, each buoy is monitored with probability 0.6. Non-monitored buoys do not generate contacts; i.e., non-monitored buoys are not counted as negative information. Each buoy scan has, on average, ten omni-directional contacts, at most one of which may be a target detection. The time difference of arrival (TDOA) errors are Gaussian with σ_{TDOA} = 3.3 seconds. The probability of detection, p^D, for each buoy is 0.4.

P-3 Radar model. The radar onboard each P-3 aircraft assumes an aircraft altitude of 3,000 feet, no range detection limit within the search area (>85 nmi), and 360° scans. Each P-3 travels six grid cells between LRT updates (two minutes), and radar data is sent to LRT after each step. Each contact consists of slant range and azimuth with Gaussian errors of σ_{Slant} = 0.54 nmi and σ_{Azimuth} = 2°, respectively. Each radar scan has, on average, twenty contacts, at most one of which may be a target detection. The probability of detection, p^D, for each radar scan is 0.4

Figure 7–6 shows the joint LRT/Air Search Planning graphical display that provides a real-time picture of the probability map that the P-3s use for planning (left figure) and the likelihood ratio surface from LRT (right figure). The two displays are synchronized in time, and each depicts the locations of the P-3s (in this case, there are two of them) and the ground truth target. The LRT display also includes the layout of the sonobuoy field.

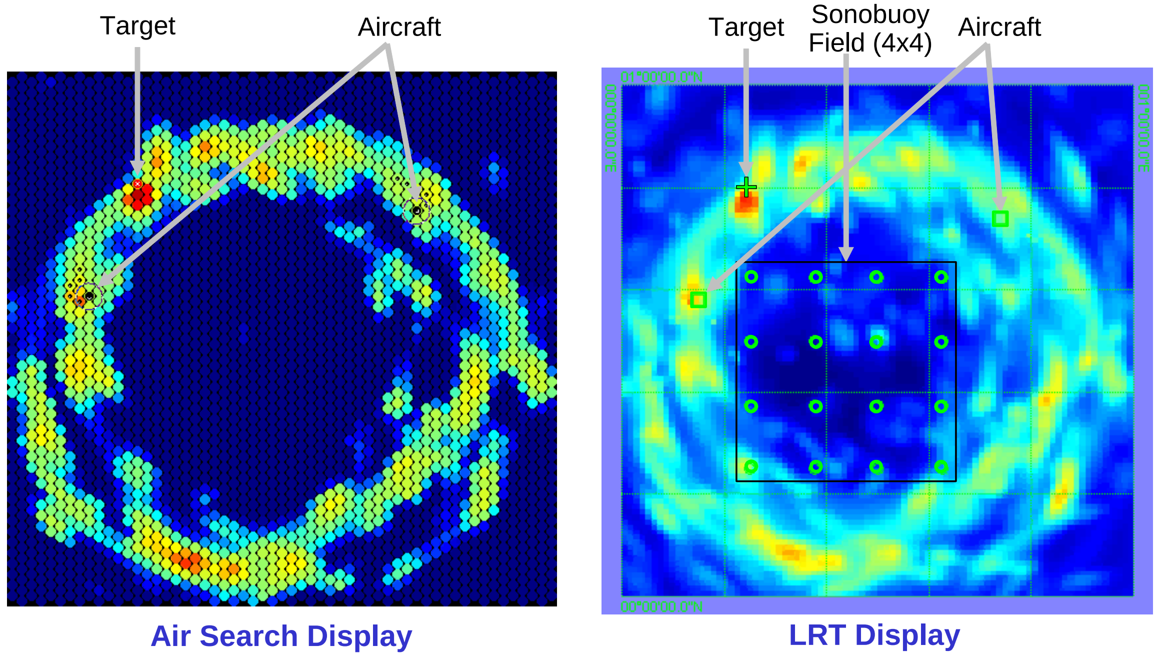


Figure 7–6: Joint LRT/Coordinated air search planning graphical display

For each set of experiments, we vary the number of P-3 search aircraft (one, two or three P-3s) and vary the amount of sensor information that is processed by LRT and used by the path planning algorithms. For the early Phase II experiments, we used a five-step path planner without dynamic sectoring and a value potential planner with sectoring. For each case, we record the target detection time over 300 independent trials. We also investigate cases in which only the Buoy information is processed, only the Radar information is processed, or both the Buoy and Radar information are processed.

Figure 7–7 shows the median time to target detection, along with the 25th and 75th percentiles, for each search path planning approach and for different numbers of P-3s. Clearly, as the amount of sensor information increases, the target detection time decreases. Furthermore, adding extra P-3 aircraft decreases target detection time for all of the test cases, as expected.

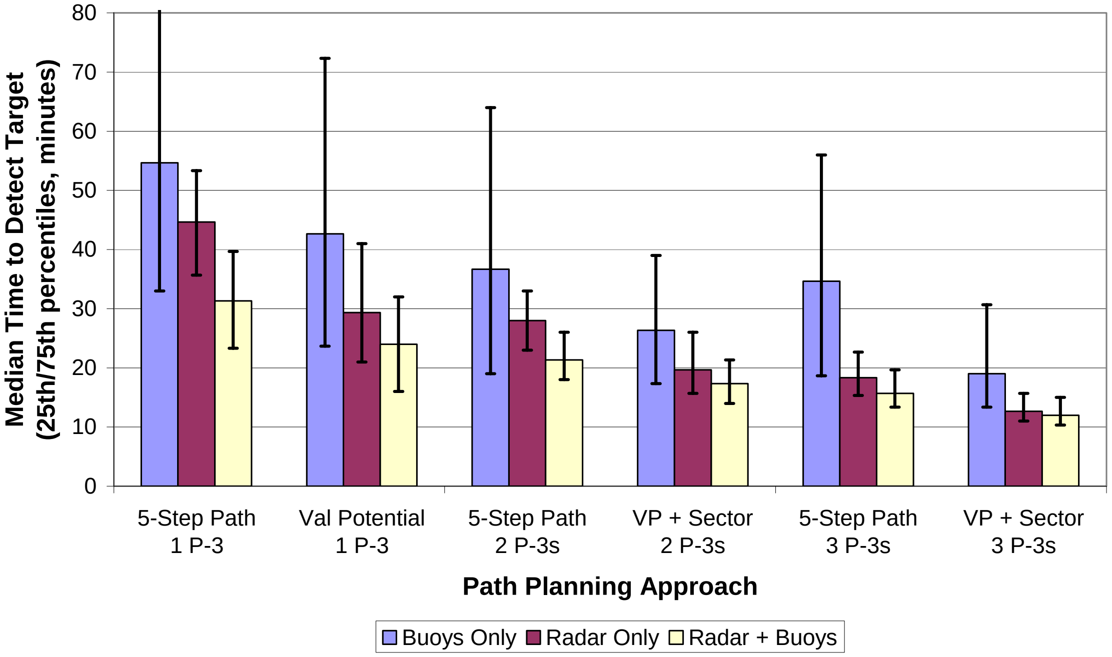


Figure 7–7: Simulation results for combining Sonobuoy and Radar sensor data

Under the Phase I research, we integrated the finite-horizon planner from the DARPA TASK work with LRT. One of the early goals under the Phase II effort was to integrate the value potential plus dynamic sectoring technologies developed under the DARPA DSO seedling effort into the LRT/Air search testbed, and then show empirically the search performance improvement early in the Phase II process. Figure 7–7 shows that these improvements were realized because the value potential planning plus dynamic sectoring reduces the detection time significantly for the different sensor information types and as the number of search aircraft increases.

It is important to reiterate the main points of the SBIR research results. Obviously, if P-3 radar scans are the only sensors available, then adding more P-3s to the search effort should decrease the target detection time. Similarly, if a sonobuoy field is the only set of sensors available, then adding more sonobuoys to the field should decrease the target detection time. However, this research shows a methodology by which the contact data from multiple, and very different, types of sensors can be fused together in real time to decrease the target detection time. Although notional sensor models were used in the Phase I and early Phase II experimental analyses, we plan to demonstrate later in the Phase II research that these benefits can be extended to real-world sensor models and contact data.

7.3. CONCLUSIONS

Under the DARPA TASK program, Metron developed and implemented technology that enables autonomous, competitive agents to negotiate the fair division of resources and tasks over time. We designed a series of agent protocols that allow the agents to perform this negotiation, and applied these protocols to two military domains: (1) commercial airlift procurement for large contingencies and (2) unmanned aerial vehicle (UAV) coordinated search and surveillance.

For the strategic airlift problem, we developed a collaborative auction and mission exchange approach that makes planning more flexible, missions more reliable, and leverages commercial operational “best practices” without having to integrate those practices into military systems or to make the expertise available to its commercial competitors. Experimental results show that this multi-agent auction plus swapping approach *cuts in half* the controllable operating cost and opportunity cost compared with the centralized assignment used today on a Gulf War-sized airlift scenario.

In the UAV domain, the challenge is achieving real-time, effective coordination among a fleet of autonomous UAVs performing intelligence, surveillance and reconnaissance (ISR) tasks. We focused our efforts on target search (detection) and surveillance (location monitoring) tasks. The developed technologies demonstrate how UAVs can plan missions collaboratively and re-plan adaptively based on real-time changes in UAV availability, pop-up targets and sensor capabilities.

For the target surveillance problem, we developed dynamic target swapping protocols, where the criterion for swapping can be greedy or cooperative and where the amount of information shared by UAVs can be relatively high or low. These swapping protocols lead to compact UAV tours that partition the space naturally from the trading behavior of the locally optimizing UAVs. In addition, we show how cooperative UAV behavior (adherence to system goals rather than strictly local goals) and greater information sharing can improve the rate of convergence to good system solutions.

For the target search problem, we developed a distributed, Bayesian tracking approach by which UAVs collaboratively plan search paths to detect mobile targets given probability distributions on target locations and estimated motion models. Each UAV optimizes its local search path, deconflicts with the search paths of the other UAVs, and shares information about where the UAV has searched and what has been sensed. Experimental results suggest that in some settings, a fleet of coordinated UAVs using this distributed approach can perform better target detection than a fleet that is three times as large following a standard lawnmower search pattern.

REFERENCES

[Axe94] R. M. Axelrod, *The Evolution of Cooperation*, Basic Books, Inc.: New York, 1994.

[Axe97] R. M. Axelrod, *The Complexity of Cooperation: Agent-Based Models of Computation*, Princeton University Press: Princeton, NJ, 1997.

[BHH59] J. Beardwood, J. H. Halton, and J. M. Hammersley, "The shortest path through many points," *Proceedings of the Cambridge Philosophical Society,* **55**, 299–327, 1959.

[BT96] S. J. Brams and A. D. Taylor, *Fair Division: from cake-cutting to dispute resolution*, Cambridge University Press, 1996

[BV97] K. Binmore and N. Vulkan, "Applying Game Theory to Automated Negotiation," *DIMACS Workshop on Economics, Game Theory and the Internet*, Rutgers University, April 1997.

[CHOSTW01] K. Chang, K. Howard, R. Oiesen, L. Shisler, M. Tanino and M. Wambsganss, "Enhancements to the FAA Ground-Delay Program under Collaborative Decision Making", *Interfaces*, **31**(1): 57–76, Jan–Feb 2001.

[FLS99] Y. Fujishima, K. Leyton-Brown and Y. Shoham. "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches," in *Proceedings of. the 16th International Joint Conference on Artificial Intelligence (IJCAI),* 548–553, 1999.

[FSJ98] P. Faratin, C. Sierra, and N.R. Jennings, "Negotiation Decision Functions for Autonomous Agents," *International Journal of Robotics and Autonomous Systems*, **24**(3-4):159–182, 1998.

[GCK05] G. Godfrey, J. Cunningham and A. Knutsen, "Negotiation Mechanisms for Coordinated Unmanned Aerial Vehicle Surveillance," *IEEE Conference on Knowledge Intensive Multiagent Systems (KIMAS 2005)*, 18-21 April 2005.

[GHK04] G. Godfrey, C. Hellings and A. Knutsen, "A Multi-Agent Framework for Collaborative Airlift Planning using Commercial Air Assets," *Mathematical and Computer Modelling*, **39**(6–8): 885–896, 2004.

[GM01] G. Godfrey and T. Mifflin, "Virtual Transportation Company REF White Paper," DARPA Taskable Agent Software Kit (TASK) program, Metron, Inc., March 2001.

[God01] G. Godfrey, "Optimization Technology for Improving Supply Chain Execution," White paper prepared for Defense Information Systems Agency, November 2001.

[God04] G. Godfrey, "Distributed, Multi-Sensor Search Optimization based on Likelihood Ratio Surfaces," Phase I SBIR Final Report for Contract N00014-04-C-0056, 30 November 2004 (Unclassified).

[God05] G. Godfrey, "Top-Down Mechanism Design Study for Multi-UAV Search and Surveillance," Final Technical Report for DARPA/ARO Contract W911NF-04-C-0041, 2005.

[Gol89] D. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Inc., 1989.

[Hug95] B. Hughes, *Random walks and random environments, Vol. 1, Random Walks*, Oxford University Press, 1995.

[JMR96] D. S. Johnson, L. A. McGeoch, and E. E. Rothberg, "Asymptotic experimental analysis for the Held-Karp traveling salesman bound," in *Proceedings 7th ACM SIAM Symposium on Discrete Algorithms*, 1996.

[JM97] D. S. Johnson and L. A. McGeoch, "The Traveling Salesman Problem," Chapter in *Local Search in Combinatorial Optimization*, E. H. L. Aarts and J. K. Lenstra (eds.), John Wiley and Sons, London, 215–310, 1997.

[JSW98] N. R. Jennings, K. Sycara and M. Wooldridge, "A Roadmap of Agent Research and Development", *Int. Journal of Autonomous Agents and Multi-Agent Systems*, **1**(1):7–38, 1998.

[JW01] N. R. Jennings and M. Wooldridge, "Agent-Oriented Software Engineering," Chapter in *Handbook of Agent Technology*, J. Bradshaw (ed.), AAAI/MIT Press, 2001.

[KL95] S. Kraus and D. Lehmann, "Designing and building a negotiating automated agent," *Computational Intelligence*, **11**(1):132–171, 1995.

[Kra97] S. Kraus, "Negotiation and cooperation in multi-agent environments," *Artificial Intelligence Journal, Special Issue on Economic Principles of Multi-Agent Systems*, **94**(1-2):79–98, 1997.

[May03] M. Maynard, "Pentagon's War Needs are a Lifeline for Airlines", *New York Times*, 26 December 2003.

[Mit96] M. Mitchell. *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1996.

[MSJ98] N. Matos, C. Sierra and N. R. Jennings, "Determining successful negotiation strategies: an evolutionary approach," *Proc. of 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, Paris, France, 182–189, 1998.

[Nwa96] H. S. Nwana, "Software Agents: An Overview," *Knowledge Engineering Review*, **11**(3):205-244, October/November 1996.

[Rob99] Statement by General Charles T. Robertson, Jr., USAF Commander in Chief, United States Transportation Command to the Senate Armed Services Committee, 10 March 1999.

[RZ94] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, MIT Press: Cambridge, MA, 1994.

[SBC99] L. Stone, C. Barlow and T. Corwin, *Bayesian Multiple Target Tracking*, Artech House, Boston, 1999.

[Ser96] F. Seredynski, "Coevolutionary Game-Theoretic Multi-Agent Systems: the Application to Mapping and Scheduling Problems," *Journal of Parallel and Distributed Computing*, **47**(1):39–57, November 1997.

[Sin84] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, **68**(2):159, 1984.

[Tum87] J. Tuma, *Engineering Mathematics Handbook, 3rd ed*, McGraw-Hill, 1987.

[Vic61] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance,* **16**, 8–27, 1961.

[VJ00] N. Vulkan and N. R. Jennings, "Efficient Mechanisms for the Supply of Services in Multi-Agent Environments," *Int. Journal of Decision Support Systems*, **28**(1-2):5–19, 2000.

[USAF98] United States Air Force, "Air Mobility Planning Factors," *Air Force Pamphlet 10-1403*, 1 March 1998.

[Wam97] M. Wambsganss, "Collaborative decision making through dynamic information transfer," *Air Traffic Control Quarterly,* **4**(2), 107–123, 1997.

[Woo97] M. Wooldridge, "Agent-based Software Engineering," *IEEE Proceedings on Software Engineering,* **144**(1):26–37, 1997.

[WJ99] M. Wooldridge and N. R. Jennings, "The Cooperative Problem Solving Process," *Journal of Logic and Computation*, **9**(4):563–592, 1999.

APPENDIX A

A. VIRTUAL TRANSPORTATION COMPANY DATA SET DESCRIPTION

A.1 Acronyms for Airlift Domain

CINC	Commander-in-Chief
CRAF	Civil Reserve Air Fleet
DOD	Department of Defense
EAD	Earliest Arrival Date
LAD	Latest Arrival Date
MSP	Maritime Security Program
POD	Point of Debarkation
POE	Point of Embarkation
RDD	Required Delivery Date
REF	Research Exploration Framework
RLD	Ready-to-Load Date
TPFDD	Time-Phased Force Deployment Database
USTRANSCOM	United States Transportation Command
VISA	Voluntary Intermodal Sealift Agreement
VTA	Voluntary Tanker Agreement
VTC	Virtual Transportation Company

A.2 MILITARY DEMAND DATABASE FIELDS

The following fields describe the military demand database:

FIELD NAME	TYPE	DESCRIPTION
Movement ID	String	Label that identifies movement requirement
Scenario 1 Day	Integer	Day on which scenario 1 is “announced”
PAX1	Integer	Number of passengers under “scenario 1” (S1 for short)
Bulk1	Float	Number of short tons of bulk cargo under S1
Oversize1	Float	Number of short tons of oversize cargo under S1
Outsize1	Float	Number of short tons of outsize cargo under S1
Origin1	String	Origin label under S1
POE1	String	Point of Embarkation label under S1
POD1	String	Point of Debarkation label under S1
Destination1	String	Destination label under S1
Orig-POE Mode1	Character	Transportation mode between Origin and POE under S1 (‘A’ for air, ‘S’ for ship, ‘L’ for land, ‘X’ for any)
POE-POD Mode1	Character	Transportation mode between POE and POD under S1
POD-Dest Mode1	Character	Transportation mode between POD and Destination under S1
RLD1	Integer	Ready-to-Load date at Origin under S1
EAD1	Integer	Earliest Arrival Date at POD under S1
LAD1	Integer	Latest Arrival Date at POD under S1
RDD1	Integer	Required delivery date at Destination under S1
Scenario 2 Day	Integer	Day to switch from scenario 1 to scenario 2
…	…	… scenario 2 contains same fields as scenario 1
RDD2	Integer	Required delivery date at Destination under S2

Table A–1: Demand Database Field Descriptions

There is no need to include all fields in all experiments. We designed the demand database to over-specify the problem with the overall philosophy that it is easier to ignore data or relax constraints for a particular experiment than to create new data. For the research community, this also means that individuals can manipulate the data according to their research interests, while keeping results comparable among different research groups when possible.

For example, for the set of airlift experiments that Metron performed, we ignored the Scenario1 data entirely. Furthermore, we ignored the Origin-POE and POD-Destination legs and focused exclusively on the long-haul POE-POD legs. Another simplification is to assume that all movement requirements are known to all parties at the beginning of the simulation (which is equivalent to all entries in the Scenario2 Day field being zero). For this experiment, then, only 10 of the 33 fields are used (Movement ID, PAX2, Bulk2, Oversize2, Outsize2, POE2, POD2, POE-POD Mode2, EAD2, LAD2). This simplified problem allowed us to debug the mission generation code without the added complications of payload uncertainty and new movement requirements that appear over time. Given that we have successfully performed some simple experimental runs to demonstrate that our code is working properly, we could include additional fields consistent with different experimental hypotheses that we want to test.

Metron also worked with individual research groups to help determine how to structure the standard data sets for the types of experiments to be performed.

A.3 TRANSPORTATION ASSET TYPES DATABASE

The following fields describe the transportation asset type database:

FIELD NAME	FIELD TYPE	DESCRIPTION
Asset ID	String	Label to identify asset type
Speed	Float	Asset speed in miles per hour
Range	Integer	Asset range (in miles) with “full tank of gas”
Mode	Character	‘A’ for air, ‘S’ for ship, ‘L’ for land
Passenger Capacity	Integer	Number of passengers that can be transported
Bulk Capacity	Float	Short tons of bulk cargo that can be transported
Oversize Capacity	Float	Short tons of oversize cargo that can be transported
Outsize Capacity	Float	Short tons of outsize cargo that can be transported
Refueling time	Integer	Minutes required to refuel asset
Turnaround time	Integer	Minutes required to load or unload at full capacity

Table A–2: Transportation Asset Type Database Field Descriptions

Initially, we have included only six asset types in the database (all of which are air assets). There are two passenger aircraft types, a commercial narrow-body and a commercial wide-body. There are four cargo aircraft types, a commercial narrow-body, a commercial wide-body, a military narrow-body, and a military wide-body. We assume that all passengers move using commercial assets, so no military passenger aircraft have been specified. Furthermore, oversized and outsized cargo can only be moved using military wide-body aircraft.

All of the speed and distance assumptions use U.S. statute miles, not nautical miles. If an aircraft is scheduled to fly a distance further than its range, then an implied enroute stop for refueling must be included in the travel time calculation. Furthermore, when an aircraft reaches the POD, separate timing charges for refueling and unloading must be included in the mission time sequentially because refueling and loading/unloading cannot be done at the same time for safety reasons.

A.4 Location Database Fields and Distance Calculation

The following fields describe the location database. The plan for future versions was to expand the ID field into a compact identifier (i.e., KHIF) and a longer label corresponding to the geographic name of the location (i.e., Hill Air Force Base). The capacity at a location is a linear combination of the number of passengers and short tons of cargo that can be processed per day. That is, Hill Air Force Base may be able to process 3000 passengers per day or 500 short tons of cargo or some linear combination of the two, such as 1500 passengers and 250 short tons of cargo in a day.

FIELD NAME	**FIELD TYPE**	**DESCRIPTION**
Location ID	String	Label to identify location
Latitude	Float	Latitude (in decimal format)
Longitude	Float	Longitude (in decimal format)
Passenger Capacity	Integer	Passengers that can be processed per day
Cargo Capacity	Integer	Cargo (short tons) that can be processed per day

Table A–3: Location Database Field Descriptions

To reduce the sensitivity of the airlift data, the locations (labels and latitude/longitude pairs) used in this data set are fictitious. In fact, most of the latitude/longitude pairs correspond to locations in the ocean. However, we have attempted to keep the distances between locations representative of a significant airlift effort from the United States to Europe/Asia/Africa.

To compute travel times on a leg, one must know the speed of the transportation asset and the length of the travel leg. The latitude and longitude of each leg endpoint are given in the included location table. Let (α_1, β_1) and (α_2, β_2) be the latitude and longitude pairs of the two locations. Given these coordinates, there are multiple formulas that may be used to compute the distance. The most well-known is the “Law of Cosines” formula, but that formula, while accurate, is ill-conditioned for short distances because of numerical precision issues when taking the inverse cosine of small values.

Instead, we will use the Haversine formula [Sin84], which is also mathematically exact, but is ill-conditioned for two points on opposite sides of the earth. However, this is less of a concern than with the Law of Cosines formula for two reasons. First, computing the distance between two points that are exactly opposite one another on the earth is uncommon. Second, the error introduced by the numerical precision issues is on the order of one mile for two points that are approximately 12,000 miles apart. The Haversine formula has two forms that rely on the same intermediate value, but uses an inverse sine or inverse tangent in the final calculation. The distance in U.S. statute (not nautical) miles, *D*, for two points with latitude/longitude pairs of (α_1, β_1) and (α_2, β_2) using the Haversine formula is:

$$c = \sin^2\left(\frac{\alpha_2 - \alpha_1}{2}\right) + \cos\alpha_1 \cos\alpha_2 \sin^2\left(\frac{\beta_2 - \beta_1}{2}\right) \tag{A.1}$$

$$D = 7912.2 \times \sin^{-1}\left(\min\left(1, \sqrt{c}\right)\right) \quad \text{or} \quad D = 7912.2 \times \tan^{-1}\left(\sqrt{1-c}, \sqrt{c}\right) \tag{A.2}$$

The “min” in the inverse sine equation is only precautionary. Mathematically, the value of *c* cannot exceed 1, but numerical precision could cause a computed value of *c* to exceed 1 by a very small amount, which would crash the inverse sine function. The inverse tangent equation does not have this limitation. Typically, the two-argument inverse tangent function is called **`atan2`** in many libraries (including Excel and Java). For the other TASK researchers developing their simulations, we suggested using the sample calculation below to make sure that the order of the arguments is correct if the **`atan2`** function is used because some libraries swap the two arguments.

To check the **`atan2`** implementation, the points (α_1, β_1) = (32°, -80°) and (α_2, β_2) = (47°, 49°) should have intermediate value *c* = 0.48821 and should be *D* = 6120.8 miles apart. To convert decimal degrees to radians, multiply the number of degrees by π/180 ≈ 0.0174533 degrees/radian.

A.5 ENTERPRISE FLEET DATABASE FIELDS

Enterprise fleet information can be stored as one file for each enterprise (which requires many files) or as a single file for all enterprises (which requires an additional field to identify the enterprise). We choose the latter format. In addition, we have defined unique identifiers for each asset type, but not for each individual asset. Had identifiers for individual assets become necessary, then we would have supported the additional field (which would have increased greatly the number of records in the fleet database). However, we concluded that tracking each individual asset would be an unproductive burden. The following fields describe the enterprise fleet database:

FIELD NAME	**FIELD TYPE**	**DESCRIPTION**
Enterprise ID	String	Label to identify enterprise
Location	String	Location code indicating where assets are located (a hub)
Asset ID	String	Label to identify asset type at that location
Quantity	Integer	Number of assets of that type at that hub for that enterprise
Opportunity Cost	Free Form	Multiple fields used to express the opportunity cost function associated with assets at the enterprise hub (see A.6)

Table A–4: Enterprise Fleet Database Field Descriptions

In general, the database lists the number of assets of each type at a particular location for a particular enterprise. The opportunity cost function (derived from the available asset inventory as described in Section 2.7) is a set of four piecewise-linear functions, one for midnight-6am, one for 6am-noon, one for noon-6pm, and one for 6pm-midnight. Each piecewise-linear function contains up to four (breakpoint, projected slope) pairs. We will describe the format and interpretation of these functions in the next section.

A.6 ECONOMIC MODEL

This appendix includes cost expressions representing DOD revenue rates, enterprise operating cost parameters, and enterprise opportunity cost functions. Three elements affect the profit that an enterprise receives when performing a military mission: (1) the revenue received, (2) the operating cost (fuel, maintenance, crew, etc.) incurred, and (3) the lost opportunity cost (what the asset could have earned commercially during the military use).

The revenue and operating cost rates are listed in Table A–5. Only the six asset types used in the fleet database are included. The revenue is expressed in dollars per passenger-mile or short ton-mile. That is, the military pays based on how many seats or how much space is needed and the round trip distance from the POE to the POD and back to the POE (for example). The operating cost, however, is strictly a function of distance (we have ignored the fact that a fully loaded aircraft consumes more fuel than an empty aircraft). Although the revenue is paid based on POE to POD to POE distance, the operating cost is incurred based on hub location to POE to POD to hub location distance. For example, if an aircraft must fly from Denver (hub) to pick up a military payload in New Jersey (POE) to deliver to Germany (POD), then the military only pays for the New Jersey-Germany-New Jersey distance, but the aircraft incurs cost for the Denver-New Jersey-Germany-Denver distance. Consequently, it is important for the enterprise to assign assets that are near the military pick up point when possible.

ASSET TYPE	**REVENUE PER PAX-MILE OR SHORT TON-MILE**	**OPERATING COST PER MILE**
Commercial Narrow-Body PAX	$0.0854	$9.09
Commercial Wide-Body PAX	$0.0672	$13.07
Commercial Narrow-Body Cargo	$0.2725	$9.07
Commercial Wide-Body Cargo	$0.2725	$15.29
Military Narrow-Body Cargo	$0.2725	$9.07
Military Wide-Body Cargo	$0.2725	$15.29

Table A–5: Revenue and Operating Cost rates by aircraft type

Although fixed revenue rates are provided, researchers may want to conduct experiments that investigate the effects of free-market pricing of the military missions. This variation is certainly allowable.

The third element needed to measure the profit of a specific mission is the opportunity cost. As mentioned in the prior appendix, the opportunity cost function for each enterprise for each location and for each asset type is described by a series of four piecewise-linear functions. Each function represents a six-hour period and is specified by up to four (breakpoint, projected slope) pairs. For example, consider the following opportunity cost database record (this appears after the enterprise, location, asset type and quantity fields):

3 0 0 2 6374 3 863 **2** 0 0 5 1539 **2** 0 0 4 1701 **2** 0 0 2 1195

Figure A–1: Opportunity Cost database record

This record describes four functions where the first function (midnight-6am) has three (breakpoint, projected slope) pairs and has marginal opportunity cost of 0, 0, 6374, 7237, 8100, 8963 and 9826 for the first seven assets used, respectively. The second function (6am-noon) has two pairs and has marginal opportunity cost of 0, 0, 0, 0, 0, 1539 and 3078 for the first seven assets used. The third function (noon-6pm) has two pairs and has marginal opportunity cost of 0, 0, 0, 0, 1701, 3402 and 5103 for the first seven assets used. Finally, the fourth function (6pm-midnight) has two pairs and has marginal opportunity cost of 0, 0, 1195, 2390, 3585, 4780 and 5975 for the first seven assets used.

While the opportunity cost function may appear cumbersome, it is an effective way of isolating the logistics details from the more important economics of the problem. For example, the agent representing an enterprise will need to compute frequently the potential profit associated with a particular military mission. Typically, the revenue is fixed and the operating cost depends on the original location of the asset used to satisfy the mission. The opportunity cost, however, is where all of the operational details are buried. Realistically, an enterprise would need to look at its entire fleet and try to identify the individual asset and specific time to satisfy the mission that maximizes the enterprise profit. In fact, commercial carriers spend hundreds of millions of dollars each year to build, maintain and use decision-support systems to help them optimize the use of their fleet. Instead of trying to reproduce these systems (too expensive) or optimizing poorly (too sloppy), we have built the opportunity cost functions to speed the process of estimating profit.

When presented with a particular mission, an enterprise can quickly determine the set of asset types compatible with that mission. The enterprise can then loop over all locations with that type of asset, computing opportunity costs at each location for each type that is consistent with how many assets are already assigned. For example, Gamma Airlines identifies that its commercial wide-body passenger (CWPAX) aircraft is suitable for a particular mission from New Jersey to Germany. Gamma's Chicago hub stations 17 CWPAX aircraft. If Gamma assigns a Chicago CWPAX aircraft to that mission, then the aircraft will leave Chicago at 6am on day 5, and return to Chicago at 9pm on day 7. The mission revenue is strictly based on seats and NJ-Germany-NJ distance. The operating cost is based on the Chicago-NJ-Germany-Chicago distance.

The opportunity cost, however, is more complicated. Breaking each day into four six-hour time periods, the aircraft will be gone for 11 time periods (three on day 5, four on day 6, and four on day 7). We will assume that if an aircraft is gone for any part of the period, then the costs are equivalent to being gone for the entire period. For each of those periods, Gamma has already assigned other CWPAX aircraft to military missions, call this the assigned aircraft state vector. The total opportunity cost is then the sum of the opportunity costs for each period given the number of aircraft already committed in that period. For the 6am-noon period on day 5, the proposed aircraft could be the fifth committed and have an opportunity cost of $5600. For the noon-6pm period on day 5, the proposed aircraft could be the third committed and have an opportunity cost of $3200. Once the total opportunity cost is tallied, Gamma can compute the total profit associated with using a Chicago CWPAX aircraft to satisfy the military mission starting at 6am on day 5.

Gamma Airlines would then consider how that total profit would change if the starting day/time was changed to noon on day 5 or any other feasible time that involved a different set of time period for the opportunity cost. This would give Gamma the minimum opportunity cost to service that mission with a CWPAX aircraft out of Chicago. By repeating this type of calculation across all of its locations that have CWPAX aircraft, Gamma can compute what location at what time with what aircraft type can maximize the profit associated with the given military mission. Although this is a time-intensive process for a human, it can be coded in software rather simply and without formal algorithms that require third-party optimization or modeling libraries.

A.7 REGULATORY CONTRACT AGREEMENTS

Civil Reserve Air Fleet (CRAF). The air component of large military airlifts goes through the Air Mobility Command (AMC) based at Scott AFB. To support these airlifts, the military created the Civil Reserve Air Fleet (CRAF) program. CRAF is a voluntary program in which commercial air carriers contractually agree to provide a fixed set of aircraft and crews (in three separate stages) to the military in times of need for a minimum 45-day period. In return, participating carriers become eligible to bid on peacetime business in proportion to their CRAF obligation. The CRAF contracts are negotiated on a yearly basis. Table A–6 lists the CRAF inventory for the year 2000, broken down by different segment types. For our purposes, we will focus our attention on the international segments and the national domestic segment.

Segment	Section		I	II	III
International	Long	Pax	44	126	325
		Cargo	37	96	207
	Short	Pax		13	84
		Cargo		4	4
National	Domestic	Pax			44
		Cargo			0
	Alaskan	Pax			0
		Cargo		6	6
Aeromedical Evacuation				25	59
TOTAL			81	270	729

Table A–6: CRAF inventory as of January 1, 2000

When military needs arise, the usual DOD procedure starts by requesting volunteered aircraft from the airlines. If volunteer assets are insufficient, then DOD can mandate their delivery by activating the appropriate CRAF stage. By requesting volunteers first, fewer commercial aircraft are tied up for less time (just enough to satisfy demand) than they would through CRAF activation.

This peacetime business is attractive to many carriers. However, CRAF activation can be extremely disruptive to the carrier enterprises. Some economic effects are short-term, such as having fewer aircraft available to satisfy the carrier's domestic schedule, and some are long-term, such as losing market share to a competitor who is not a CRAF participant.

Sealift Programs. The Maritime Security Program (MSP) and the Voluntary Intermodal Sealift Agreement (VISA) address sea transport. Under MSP, ten carriers have committed 47 ships to be made available to DOD in exchange for an annual retainer fee. The MSP ships are required to also enroll in VISA, in which 35 companies and 109 oceangoing dry-cargo liner vessels participate, along with a number of tugs and barges. As with CRAF, there are three stages to VISA. Operators can volunteer capacity in Stages I and II, but in Stage III, they must commit at least 50 percent of their vessel capacity. Furthermore, MSP participants must commit 100 percent of their MSP assets under Stage III VISA activation.

VISA also includes access to the intermodal transportation resources of the commercial carriers, including trains, trucks, cargo handling equipment, cargo tracking and control systems, and traffic and logistics management services.

Database Elements. For the initial data sets, we consider only CRAF participants with the following database fields:

FIELD NAME	**FIELD TYPE**	**DESCRIPTION**
Enterprise ID	String	Label to identify enterprise
Stage	Integer	Relevant CRAF stage for stated obligation
Asset Type	String	Label to identify asset type
Quantity	Integer	Number of assets of given type that the enterprise is obligated to provided under listed CRAF stage

Table A–7: CRAF Obligation Database Field descriptions

If the military needs to invoke CRAF and the airlift shortfall is between Stage I and II, then the military would invoke CRAF Stage II. However, only the proportion of Stage II assets that are needed would be called up, meaning that each carrier would provide a sufficient fraction of its Stage II assets. While this calculation is easy to do after CRAF is invoked, the carriers need to compute how much of its obligation has been satisfied via volunteered aircraft.

An appropriate way to compute this satisfied obligation is to start by computing the CRAF obligation in slightly different terms. Instead of using entire aircraft, this modified calculation simply counts seats and short ton capacity associated with each CRAF stage. For example, using these Stage I calculations, suppose Alpha Airlines has 3,000 seats, Bravo Airlines has 2,000 seats and Charlie Airlines has 5,000 seats. Then for CRAF Stage I activation, Alpha, Bravo and Charlie are responsible for 30%, 20% and 50% of the airlift, respectively. If CRAF Stage II activation was needed, then these proportions would reflect the Stage II seat capacities for each carrier.

Having converted the raw aircraft obligation into a proportionate airlift obligation, we can compute the proportion of airlift satisfied by an enterprise at any time similarly. The calculation can be done in two ways. One approach takes all of the demand requirements and converts them into total passenger-miles or short ton-miles by computing the POE-POD-POE round trip distance and multiplying by the number of passengers or short tons of cargo. Multiplying this aggregate lift total by each enterprise's obligation under the relevant CRAF stage produces an aggregate lift that each enterprise is required to perform. For each mission assigned to the enterprise, the appropriate passenger-mile or short ton-mile total can be subtracted from the enterprise's obligation.

Another approach computes the aggregate passenger-mile or short ton-mile total only for missions that have been already assigned. By taking the ratio of aggregate lift assigned to a particular enterprise to aggregate lift assigned to all enterprises, the enterprise can compare its fraction of lift fulfilled with the fraction of lift it is obligated to perform.

Either calculation can be performed as the airlift assignment progresses. Once the entire airlift has been assigned, the two calculations will yield the same values. More sophisticated variations will be considered in the future, such as fulfilling the obligation over each seven-day period, rather than over the entire airlift. Other notions of measuring fairness may also be considered.

APPENDIX B

B. SHAKE OUT ALGORITHM FOR AIRCRAFT AVAILABILITY

The ideas in this appendix were based on discussions with Mr. Roger Beatty, Air Operations Specialist for American Airlines, and Dr. Michael Finn, a Metron Senior Analyst. In this appendix, we describe how to rearrange the aircraft assigned to departure flights in order to "shake out" an aircraft for CRAF use during a specific time interval.

Each arrival has a corresponding departure. By permuting the arrivals assigned to each departure (and adding delays, if any), the goal is to find a new schedule that allows the departure and arrival of a CRAF mission within acceptable delay to the existing schedule. Several optimal permutations may minimize delay. Select the permutation that minimizes the disruption to the original schedule; that is, the number of altered connections.

B.1 SHAKE OUT ALGORITHM

Assume that AMC is requesting an aircraft from time t_1 to t_2, which is represented by a departure at time t_1 and an arrival at t_2 that cannot be delayed. The permutation π rearranges the arrival associated with each departure.

The shake out algorithm works by rolling time forward and modifying the permutation π with pair-wise switches in order to meet the demand (departing flights) at that time. We start at time t_1, which is the earliest (and only) unfilled demand in the original schedule. We look at the planes on the ground that are ready to go at time t_1. From that group, we choose to send the plane with the latest departure time to the AMC; this is a pair-wise switch in π. If no plane is available, then delay the latest departure prior to t_1 and assign that plane to CRAF. This procedure shakes out the aircraft that reports to CRAF.

For subsequent unassigned departures, the procedure is similar: assign the departure to the available aircraft on the ground with the latest scheduled departure. Since non-CRAF departures can be delayed, if no planes are available, then the departure is delayed until the first arriving plane becomes available.

B.1.1 Algorithm Description

1. Assign aircraft to CRAF.
    a. List available aircraft at time of CRAF departure. Is this set empty?
    b. If the set is not empty, then assign the aircraft with the latest scheduled departure time to CRAF.
    c. If the set is empty, take the aircraft with the latest scheduled departure prior to the CRAF departure and use it for the CRAF.
2. Find aircraft for unassigned departures.
    a. Find the earliest unassigned departure. Can the returning CRAF plane satisfy the departure?
    b. If so, then assign that aircraft to the departure, and terminate.
    c. If not, then list all available aircraft on the ground at the departure time. Is this set empty?
    d. If this set is not empty, then assign the aircraft with the latest scheduled departure time to the unassigned departure. Go to step 2a.
    e. If this set is empty, then delay the departure until the first arriving aircraft becomes available, including the aircraft returning from CRAF. If the CRAF aircraft is used, then terminate. Otherwise, go to step 2a.

B.1.2 Application of Shake Out Algorithm to JFK Airport Test Data

We illustrate the algorithm with an example in which AMC requests an aircraft from JFK Airport from 0600 on August 12 to 0600 on August 13. Figure B–1 shows the resulting schedule changes.

ID	Arrival	Time	Departure	Time
			CRAF	***8/12 0600***
1	**1600/11**	**8/11 2141**	1735/12	8/12 0645
2	**656/11**	**8/11 1956**	587/12	8/12 0700
3	**1498/11**	**8/11 2158**	699/12	8/12 0720
4	**658/11**	**8/11 2036**	669/12	8/12 0820
5	**688/11**	**8/11 2047**	735/12	8/12 0855
6	**141/11**	**8/11 2311**	106/12	8/12 0930
7	**9572/11**	**8/11 1828**	611/12	8/12 0930
8	**670/11**	**8/11 1741**	645/12	8/12 1000
9	518/12	8/12 1001	657/12	8/12 1125
10	***1290/11***	***8/12 0042***	1473/12	8/12 1130
11	662/12	8/12 1049	1819/12	8/12 1205
12	816/12	8/12 1024	635/12	8/12 1230
13	648/12	8/12 1425	647/12	8/12 1525
14	588/12	8/12 1610	1459/12	8/12 1830
15	670/12	8/12 1718	1169/12	8/12 1900
16	664/12	8/12 1830	132/12	8/12 2115
17	1600/12	8/12 2015	899/12	8/12 2145
18	1416/12	8/12 1648	663/12	8/12 2355
19	638/12	8/12 2204	1735/13	8/13 0645
20	658/12	8/12 2035	587/13	8/13 0700
21	656/12	8/12 1936	699/13	8/13 0720
22	1498/12	8/12 2220	669/13	8/13 0820
23	1290/12	8/13 0045	735/13	8/13 0855
24	101/12	8/12 1251	611/13	8/13 0930
25	141/12	8/12 2320	106/13	8/13 0930
26	688/12	8/12 2123	645/13	8/13 1000
27	1728/12	8/12 1807	1473/13	8/13 1130
28	CRAF	8/13 0600		

ID	Arrival	Time	Departure	Time
10	1290/11	8/12 0042	CRAF	8/12 0600
1	1600/11	8/11 2141	1735/12	8/12 0645
2	656/11	8/11 1956	587/12	8/12 0700
3	1498/11	8/11 2158	699/12	8/12 0720
4	658/11	8/11 2036	669/12	8/12 0820
5	688/11	8/11 2047	735/12	8/12 0855
6	141/11	8/11 2311	106/12	8/12 0930
7	9572/11	8/11 1828	611/12	8/12 0930
8	670/11	8/11 1741	645/12	8/12 1000
9	518/12	8/12 1001	657/12	8/12 1125
			1473/12	***8/12 1130***
11	**662/12**	**8/12 1049**	1819/12	8/12 1205
12	***816/12***	***8/12 1024***	635/12	8/12 1230
13	648/12	8/12 1425	647/12	8/12 1525
14	588/12	8/12 1610	1459/12	8/12 1830
15	670/12	8/12 1718	1169/12	8/12 1900
16	664/12	8/12 1830	132/12	8/12 2115
17	1600/12	8/12 2015	899/12	8/12 2145
18	1416/12	8/12 1648	663/12	8/12 2355
19	638/12	8/12 2204	1735/13	8/13 0645
20	658/12	8/12 2035	587/13	8/13 0700
21	656/12	8/12 1936	699/13	8/13 0720
22	1498/12	8/12 2220	669/13	8/13 0820
23	1290/12	8/13 0045	735/13	8/13 0855
24	101/12	8/12 1251	611/13	8/13 0930
25	141/12	8/12 2320	106/13	8/13 0930
26	688/12	8/12 2123	645/13	8/13 1000
27	1728/12	8/12 1807	1473/13	8/13 1130
28	CRAF	8/13 0600		

ID	Arrival	Time	Departure	Time
10	1290/11	8/12 0042	CRAF	8/12 0600
1	1600/11	8/11 2141	1735/12	8/12 0645
2	656/11	8/11 1956	587/12	8/12 0700
3	1498/11	8/11 2158	699/12	8/12 0720
4	658/11	8/11 2036	669/12	8/12 0820
5	688/11	8/11 2047	735/12	8/12 0855
6	141/11	8/11 2311	106/12	8/12 0930
7	9572/11	8/11 1828	611/12	8/12 0930
8	670/11	8/11 1741	645/12	8/12 1000
9	518/12	8/12 1001	657/12	8/12 1125
12	816/12	8/12 1024	1473/12	8/12 1130
11	662/12	8/12 1049	1819/12	8/12 1205
			635/12	***8/12 1230***
13	648/12	8/12 1425	647/12	8/12 1525
14	588/12	8/12 1610	1459/12	8/12 1830
15	670/12	8/12 1718	1169/12	8/12 1900
16	664/12	8/12 1830	132/12	8/12 2115
17	1600/12	8/12 2015	899/12	8/12 2145
18	1416/12	8/12 1648	663/12	8/12 2355
19	638/12	8/12 2204	1735/13	8/13 0645
20	658/12	8/12 2035	587/13	8/13 0700
21	656/12	8/12 1936	699/13	8/13 0720
22	1498/12	8/12 2220	669/13	8/13 0820
23	1290/12	8/13 0045	735/13	8/13 0855
24	***101/12***	***8/12 1251***	611/13	8/13 0930
25	141/12	8/12 2320	106/13	8/13 0930
26	688/12	8/12 2123	645/13	8/13 1000
27	1728/12	8/12 1807	1473/13	8/13 1130
28	CRAF	8/13 0600		

ID	Arrival	Time	Departure	Time
10	1290/11	8/12 0042	CRAF	8/12 0600
1	1600/11	8/11 2141	1735/12	8/12 0645
2	656/11	8/11 1956	587/12	8/12 0700
3	1498/11	8/11 2158	699/12	8/12 0720
4	658/11	8/11 2036	669/12	8/12 0820
5	688/11	8/11 2047	735/12	8/12 0855
6	141/11	8/11 2311	106/12	8/12 0930
7	9572/11	8/11 1828	611/12	8/12 0930
8	670/11	8/11 1741	645/12	8/12 1000
9	518/12	8/12 1001	657/12	8/12 1125
12	816/12	8/12 1024	1473/12	8/12 1130
11	662/12	8/12 1049	1819/12	8/12 1205
24	101/12	8/12 1251	***635/12***	***8/12 1321***
13	648/12	8/12 1425	647/12	8/12 1525
14	588/12	8/12 1610	1459/12	8/12 1830
15	670/12	8/12 1718	1169/12	8/12 1900
16	664/12	8/12 1830	132/12	8/12 2115
17	1600/12	8/12 2015	899/12	8/12 2145
18	1416/12	8/12 1648	663/12	8/12 2355
19	638/12	8/12 2204	1735/13	8/13 0645
20	658/12	8/12 2035	587/13	8/13 0700
21	656/12	8/12 1936	699/13	8/13 0720
22	1498/12	8/12 2220	669/13	8/13 0820
23	1290/12	8/13 0045	735/13	8/13 0855
			611/13	***8/13 0930***
25	141/12	8/12 2320	106/13	8/13 0930
26	688/12	8/12 2123	645/13	8/13 1000
27	1728/12	8/12 1807	1473/13	8/13 1130
28	***CRAF***	***8/13 0600***		

Figure B–1. Modification of JFK Schedule to Add CRAF Assignment

In the upper-left part of Figure B–1, we have the original schedule along with an extra departure at 0600 on 8/12 and an extra arrival at 0600 on 8/13. To accommodate the departure (labeled CRAF), we list all arrivals that are available at 0600 (shaded in blue) and pick the one that has the latest departure time, #1290/11. This aircraft is now assigned to the CRAF mission, but we have to find an aircraft to satisfy the abandoned departure #1473/12 at 1130.

In the upper-right part of Figure B–1, we continue the algorithm as before, identifying arrivals that can satisfy the unassigned departure #1473/12 and selecting the one that has the latest departure time (#816/12). This process continues to the lower-left part of Figure B–1. In this case, there are no aircraft available at 1230 to cover #635/12, so we select the next arriving flight (#101/12 at 1251) and delay the departure until 1321 to allow the arrival at 1251 to be serviced.

Finally, in the lower-right part of Figure B–1, the unassigned departure (#611/13 at 0930) can be satisfied by the CRAF flight arriving at 0600, so the assignment is made and the algorithm is finished.

B.2 GENERALIZATION TO ESTIMATE AVAILABLE CAPACITY

The shake out algorithm was designed to modify a schedule in order to handle a request for a single aircraft over a specific period. For planning purposes, though, it would be helpful to know how much capacity (quantity and duration) can be "shaken out" of a schedule within specific delay guidelines. It would not be necessary, in the planning phase, to compute the permutation necessary to add this estimated capacity into the existing schedule.

B.2.1 Basic Idea

The basic idea behind this capacity estimation algorithm is that any aircraft on the ground is available for CRAF usage as long as it returns (and is serviced) before its next departure flight. The permutations in the original shake out algorithm allow other aircraft on the ground to act as substitutes for departures assigned to the CRAF aircraft. If we ignore the permutations and treat all arriving aircraft as identical, we can estimate the CRAF capacity by tracking the inventory of available aircraft on the ground at any time.

Figure B–2 illustrates the inventory for the JFK schedule in the previous section. After an aircraft arrives, we add a 30-minute delay for service and then increase the on-ground, available inventory by one. When an aircraft departs, we decrease the inventory by one. By viewing the inventory over time, the CRAF capacity can be identified by continuous blocks of time in which *x* aircraft are available.

For example, Figure B–2 shows that a single aircraft is available from 1811 on 8/11 to 1230 on 8/12. CRAF could use this aircraft from 1811 on 8/11 until 1200 on 8/12, when it would be serviced to be available for the 1230 departure. In addition, another aircraft would also be available from 1321 on 8/12 to 1130 on 8/13.

If we allow flight #635/12 that departs at 1230 on 8/12 to be delayed until 1321, then CRAF could use the aircraft from 1811 on 8/11 until 1100 on 8/13, a total of 41 hours. The airline would incur only a 51-minute delay to its schedule. Notice that this is exactly the same conclusion drawn from the shake out algorithm in the previous section, except with this inventory approach, we do not know how the schedule must be juggled to get this result.

Before shaking out additional capacity from the schedule, we update the existing schedule to include the CRAF mission by adding a departure on 8/11 at 1811 and an arrival on 8/13 at 1100. The 51-minute delay is also incorporated into the departure time of flight #635/12. In Figure B–3, the updated available inventory reveals that another lengthy CRAF mission can be added if two short delays and one longer delay on 8/12 is acceptable. The delays are determined by the gaps when the inventory goes to zero.

In this case, the 1000 departure must be delayed until 1031, the 1205 departure must be delayed until 1455, and the 1525 departure must be delayed until 1640. However, by accepting the delays, another aircraft can be committed to CRAF from 1858 on 8/11 until 0930 on 8/13, a total of 38.5 hours. Otherwise, the CRAF commitment would be split into two missions, one running from 1858 on 8/11 until 0930 on 8/12 and the other running from 1640 on 8/12 until 0930 on 8/13.

Additional shake outs can be performed similarly, and the length of the proposed CRAF commitments can be compared easily against the delays to the existing schedule. The advantage of this inventory approach is its ease of use, but determining the exact sequence of arrivals and departures needed to modify the schedule requires the permutation algorithm described earlier.

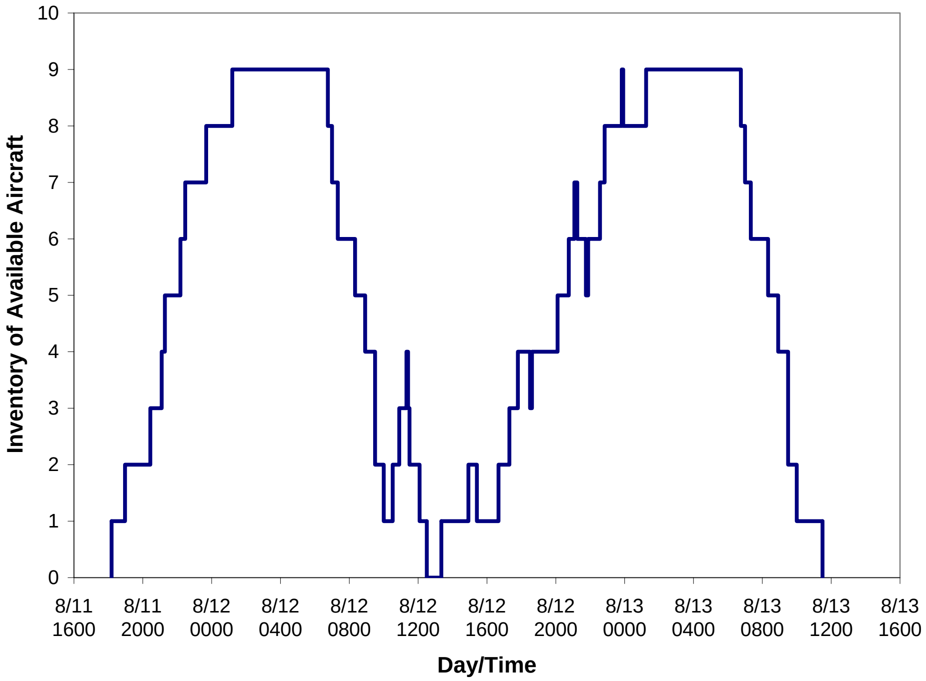


Figure B–2: Available Aircraft Inventory at JFK Airport

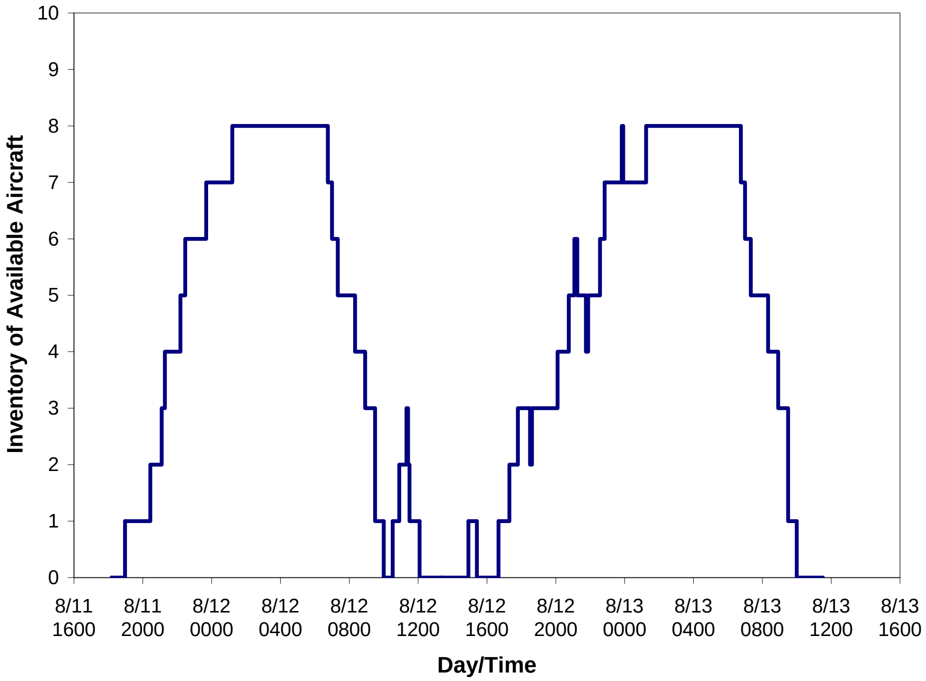


Figure B–3: Available Aircraft Inventory at JFK Airport after the first shake out

B.2.2 Automating the Inventory Approach (no delays)

In order to automate the process of extracting capacity, we need the following data:

- List of arrival times
- List of departure times
- Bin time length to discretize the timeline (say, 15 minutes)
- The following integer vectors with *N* components, indexed by *n*=0, 1, …, *N*–1:
  - The number of arrivals that become available during each bin interval, denoted Arrival[*n*]
  - The number of departures during each bin interval, denoted Departure[*n*]
  - The net inventory at the end of each bin interval, denoted NetInv[*n*]
  - A "streak" count for each bin interval that determine the length of the current streak of positive or zero inventory, denoted Streak[*n*]

The bins used in each vector track a number of times that an event occurs during the bin time interval. For our examples, we will use a bin length of one hour. For example, the first bin tracks activity between 0900 and 1000, the second bin tracks activity between 1000 and 1100, and so on. In practice, a shorter interval such as 10 or 15 minutes may be more appropriate.

The steps of the algorithm are as follows:

1. Set Arrival[*n*], Departure[*n*], NetInv[*n*], and Streak[*n*] variables to zero, for all *n* = 0, 1, …, *N*–1.
2. For each arrival, calculate the time of availability (by adding the service time to the arrival time) and find the index *n* of the bin corresponding to that time. Increment Arrival[*n*] by one.
3. For each departure, calculate the index *n* of the bin corresponding to the departure time. Increment Departure[*n*] by one.
4. Calculate the net available inventory for each bin.
   1. For *n* = 0, NetInv[*n*] = Arrival[*n*] – Departure[*n*]
   2. For *n* = 1,…,*N*–1, NetInv[*n*] = NetInv[*n*-1] + Arrival[*n*] – Departure[*n*]

5. Compute the streaks as follows

   a. Let Streak[0] = 0 if NetInv[0] > 0. Otherwise, let Streak[0] = –1.

   b. For *n* = 1, …, *N*-1, there are four possibilities:

      i) If NetInv[*n*] > 0 and Streak[*n*-1] >= 0, then let Streak[*n*] = Streak[*n*-1] + 1.

      ii) If NetInv[*n*] > 0 and Streak[*n*-1] < 0, then let Streak[*n*] = 0.

      iii) If NetInv[*n*] = 0 and Streak[*n*-1] >= 0, then let Streak[*n*] = –1.

      iv) If NetInv[*n*] = 0 and Streak[*n*-1] < 0, then let Streak[*n*] = Streak[*n*-1] – 1.

      v) Pseudo code for implementation

```
if (NetInv[n] > 0) {
        if (Streak[n-1] >= 0)  Streak[n] = Streak[n-1] + 1;
        else  Streak[n] = 0;
}
else if (Streak[n-1] >= 0)  Streak[n] = -1;
else Streak[n] = Streak[n-1] – 1;
```

   c. Adjust the streak values as follows, for *n* = *N*–2, …, 0:

      i) If Streak[*n*+1] >= 0 and Streak[*n*] >= 0, then Streak[*n*] = Streak[*n*+1].

      ii) If Streak[*n*+1]<0, Streak[*n*]<0, and Departure[*n*+1]=0, then Streak[*n*] = Streak[*n*+1].

Figure B–4 illustrates the steps listed above with the JFK Airport data for the second and third shake outs. The Streak column shows the value of the Streak[*n*] array after Step 5b and the Adjust column shows the value of the Streak[*n*] array after Step 5c.

Given the adjusted Streak[*n*] array, it is straightforward to identify excess capacity in the schedule by identifying the intervals over which Streak[*n*] > 0. For example, in the first table, there are three intervals when capacity is available. The first interval begins at 8/11 1900 and lasts for 14 periods (to 8/12 0900, and must become available by 8/12 1000). The second interval begins at 8/12 1100 and lasts for one period (to 8/12 1200). The final interval begins at 8/12 1700 and lasts for 16 periods (to 8/13 0900).

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-1
8/11 1900	2	1	1	0	14
8/11 2000	0	0	1	1	14
8/11 2100	1	0	2	2	14
8/11 2200	2	0	4	3	14
8/11 2300	2	0	6	4	14
8/12 0000	1	0	7	5	14
8/12 0100	0	0	7	6	14
8/12 0200	1	0	8	7	14
8/12 0300	0	0	8	8	14
8/12 0400	0	0	8	9	14
8/12 0500	0	0	8	10	14
8/12 0600	0	0	8	11	14
8/12 0700	0	2	6	12	14
8/12 0800	0	1	5	13	14
8/12 0900	0	2	3	14	14
8/12 1000	0	3	0	-1	-1
8/12 1100	2	0	2	0	1
8/12 1200	1	2	1	1	1
8/12 1300	0	1	0	-1	-2
8/12 1400	1	1	0	-2	-2
8/12 1500	1	0	1	0	0
8/12 1600	0	1	0	-1	-1
8/12 1700	1	0	1	0	16
8/12 1800	2	0	3	1	16
8/12 1900	2	2	3	2	16
8/12 2000	0	0	3	3	16
8/12 2100	2	0	5	4	16
8/12 2200	2	2	5	5	16
8/12 2300	2	0	7	6	16
8/13 0000	1	1	7	7	16
8/13 0100	0	0	7	8	16
8/13 0200	1	0	8	9	16
8/13 0300	0	0	8	10	16
8/13 0400	0	0	8	11	16
8/13 0500	0	0	8	12	16
8/13 0600	0	0	8	13	16
8/13 0700	0	2	6	14	16
8/13 0800	0	1	5	15	16
8/13 0900	0	2	3	16	16
8/13 1000	0	3	0	-1	-3
8/13 1100	0	0	0	-2	-3
8/13 1200	1	1	0	-3	-3

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-3
8/11 1900	2	2	0	-2	-3
8/11 2000	0	0	0	-3	-3
8/11 2100	1	0	1	0	12
8/11 2200	2	0	3	1	12
8/11 2300	2	0	5	2	12
8/12 0000	1	0	6	3	12
8/12 0100	0	0	6	4	12
8/12 0200	1	0	7	5	12
8/12 0300	0	0	7	6	12
8/12 0400	0	0	7	7	12
8/12 0500	0	0	7	8	12
8/12 0600	0	0	7	9	12
8/12 0700	0	2	5	10	12
8/12 0800	0	1	4	11	12
8/12 0900	0	2	2	12	12
8/12 1000	1	3	0	-1	-1
8/12 1100	2	0	2	0	1
8/12 1200	1	2	1	1	1
8/12 1300	0	1	0	-1	-2
8/12 1400	1	1	0	-2	-2
8/12 1500	1	0	1	0	0
8/12 1600	0	1	0	-1	-2
8/12 1700	1	1	0	-2	-2
8/12 1800	2	0	2	0	15
8/12 1900	2	2	2	1	15
8/12 2000	0	0	2	2	15
8/12 2100	2	0	4	3	15
8/12 2200	2	2	4	4	15
8/12 2300	2	0	6	5	15
8/13 0000	1	1	6	6	15
8/13 0100	0	0	6	7	15
8/13 0200	1	0	7	8	15
8/13 0300	0	0	7	9	15
8/13 0400	0	0	7	10	15
8/13 0500	0	0	7	11	15
8/13 0600	0	0	7	12	15
8/13 0700	0	2	5	13	15
8/13 0800	0	1	4	14	15
8/13 0900	0	2	2	15	15
8/13 1000	1	3	0	-1	-3
8/13 1100	0	0	0	-2	-3
8/13 1200	1	1	0	-3	-3

Figure B–4: Inventory Vectors for Second and Third Shake Outs

If only the first and last intervals are accepted for CRAF missions, the vectors above are adjusted by adding two new departures and two new arrivals corresponding to the CRAF missions. The second table in Figure B–4 shows the updated vectors once these missions have been added. The changes to the arrival and departure vectors are shaded in the figure. The process of shaking out additional capacity continues until no further reasonable CRAF assignments can be made.

B.2.3 Automating the Inventory Approach (with delays)

If delays are allowed, then the analysis becomes trickier. The algorithm is mostly the same with a change to the streak calculation to take into account whether a departure has occurred during a time period with zero inventory.

5. Compute the streaks as follows
    - b. For *n* = 1, …, *N*–1, there are five possibilities:
        - i) If NetInv[*n*] > 0 and Streak[*n*-1] >= 0, then let Streak[*n*] = Streak[*n*-1] + 1.
        - ii) If NetInv[*n*] > 0 and Streak[*n*-1] < 0, then let Streak[*n*] = 0.
        - iii) If NetInv[*n*] = 0 and Departure[*n*] > 0, then let Streak[*n*] = –1.
        - iv) If NetInv[*n*] = 0 and Departure[*n*] = 0 and Streak[*n*-1] >= 0, then let Streak[*n*] = –1.
        - v) If NetInv[*n*] = 0 and Departure[*n*] = 0 and Streak[*n*-1] < 0, then let Streak[*n*] = Streak[*n*-1] – 1.
        - vi) Pseudo code for implementation

```
if (NetInv[n] > 0) {
        if (Streak[n-1] >= 0)  Streak[n] = Streak[n-1] + 1;
        else  Streak[n] = 0;
}
else if ((Streak[n-1] >= 0) OR (Departure[n] > 0))  Streak[n] = –1;
else Streak[n] = Streak[n-1] – 1;
```

Figure B–5 shows the new streak computation applied to the JFK Airport data for the second shake out. The first table shows the streaks with no delay, and the second table shows the effect of delaying four flights for approximately one hour each. With the new streak computation, it is simple to determine how much delay is necessary to eliminate the zero inventory condition.

Whenever the inventory goes to zero, it is due to having at least one departure during that period (or due to having an initial inventory of zero). The adjusted streak value describes how many periods one of those departures must be delayed in order to reach the next departure or until an aircraft arrives to cover the departure. In many cases, it is better to delay several consecutive departures for a short period than a single departure for a long period. For example, suppose we have a 1200 departure, a 1500 departure (with an available aircraft), and a 1730 arrival. Instead of delaying the 1200 departure for six hours, you could delay the 1200 departure for three hours and use the aircraft assigned for the 1500 departure, and then delay the 1500 departure for three hours and assign it to the aircraft that becomes available at 1800.

The second table in Figure B–5 shows the effect of adding the four, one-hour delays to the departures in the shaded periods. Consequently, CRAF can have an aircraft for 38 periods instead of 30 (14+16) periods. While this may not seem significant, it could allow an international flight that was otherwise infeasible.

One drawback to this approach is that we do not identify specific flights to be delayed. This could be added with a little extra bookkeeping. In addition, there is a problem with accidentally delaying a flight twice, once each during two different shake outs. While each delay could be of a reasonable length, combining two delays may be unacceptable. This can also be avoided with appropriate bookkeeping.

Finally, we add the second CRAF aircraft by adding a departure at 8/11 1900 and an arrival at 8/13 0800 (which must be available no later than 8/13 0900). In Figure B–6, we analyze a third shake out that can create a CRAF aircraft for 36 periods while incurring three delays of one hour and two delays of two hours each.

The shake out process can continue until additional aircraft cannot be obtained without significant delays (where the level of significance depends on the application).

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-1
8/11 1900	2	1	1	0	14
8/11 2000	0	0	1	1	14
8/11 2100	1	0	2	2	14
8/11 2200	2	0	4	3	14
8/11 2300	2	0	6	4	14
8/12 0000	1	0	7	5	14
8/12 0100	0	0	7	6	14
8/12 0200	1	0	8	7	14
8/12 0300	0	0	8	8	14
8/12 0400	0	0	8	9	14
8/12 0500	0	0	8	10	14
8/12 0600	0	0	8	11	14
8/12 0700	0	2	6	12	14
8/12 0800	0	1	5	13	14
8/12 0900	0	2	3	14	14
8/12 1000	0	3	0	-1	-1
8/12 1100	2	0	2	0	1
8/12 1200	1	2	1	1	1
8/12 1300	0	1	0	-1	-1
8/12 1400	1	1	0	-1	-1
8/12 1500	1	0	1	0	0
8/12 1600	0	1	0	-1	-1
8/12 1700	1	0	1	0	16
8/12 1800	2	0	3	1	16
8/12 1900	2	2	3	2	16
8/12 2000	0	0	3	3	16
8/12 2100	2	0	5	4	16
8/12 2200	2	2	5	5	16
8/12 2300	2	0	7	6	16
8/13 0000	1	1	7	7	16
8/13 0100	0	0	7	8	16
8/13 0200	1	0	8	9	16
8/13 0300	0	0	8	10	16
8/13 0400	0	0	8	11	16
8/13 0500	0	0	8	12	16
8/13 0600	0	0	8	13	16
8/13 0700	0	2	6	14	16
8/13 0800	0	1	5	15	16
8/13 0900	0	2	3	16	16
8/13 1000	0	3	0	-1	-2
8/13 1100	0	0	0	-2	-2
8/13 1200	1	1	0	-1	-1

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-1
8/11 1900	2	1	1	0	38
8/11 2000	0	0	1	1	38
8/11 2100	1	0	2	2	38
8/11 2200	2	0	4	3	38
8/11 2300	2	0	6	4	38
8/12 0000	1	0	7	5	38
8/12 0100	0	0	7	6	38
8/12 0200	1	0	8	7	38
8/12 0300	0	0	8	8	38
8/12 0400	0	0	8	9	38
8/12 0500	0	0	8	10	38
8/12 0600	0	0	8	11	38
8/12 0700	0	2	6	12	38
8/12 0800	0	1	5	13	38
8/12 0900	0	2	3	14	38
8/12 1000	0	2	1	15	38
8/12 1100	2	1	2	16	38
8/12 1200	1	2	1	17	38
8/12 1300	0	0	1	18	38
8/12 1400	1	1	1	19	38
8/12 1500	1	1	1	20	38
8/12 1600	0	0	1	21	38
8/12 1700	1	1	1	22	38
8/12 1800	2	0	3	23	38
8/12 1900	2	2	3	24	38
8/12 2000	0	0	3	25	38
8/12 2100	2	0	5	26	38
8/12 2200	2	2	5	27	38
8/12 2300	2	0	7	28	38
8/13 0000	1	1	7	29	38
8/13 0100	0	0	7	30	38
8/13 0200	1	0	8	31	38
8/13 0300	0	0	8	32	38
8/13 0400	0	0	8	33	38
8/13 0500	0	0	8	34	38
8/13 0600	0	0	8	35	38
8/13 0700	0	2	6	36	38
8/13 0800	0	1	5	37	38
8/13 0900	0	2	3	38	38
8/13 1000	0	3	0	-1	-2
8/13 1100	0	0	0	-2	-2
8/13 1200	1	1	0	-1	-1

Figure B–5: Inventory Vectors for Second Shake Out showing the effect of delaying four flights for approximately one hour each

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-1
8/11 1900	2	2	0	-1	-2
8/11 2000	0	0	0	-2	-2
8/11 2100	1	0	1	0	12
8/11 2200	2	0	3	1	12
8/11 2300	2	0	5	2	12
8/12 0000	1	0	6	3	12
8/12 0100	0	0	6	4	12
8/12 0200	1	0	7	5	12
8/12 0300	0	0	7	6	12
8/12 0400	0	0	7	7	12
8/12 0500	0	0	7	8	12
8/12 0600	0	0	7	9	12
8/12 0700	0	2	5	10	12
8/12 0800	0	1	4	11	12
8/12 0900	0	2	2	12	12
8/12 1000	0	2	0	-1	-1
8/12 1100	2	1	1	0	0
8/12 1200	1	2	0	-1	-2
8/12 1300	0	0	0	-2	-2
8/12 1400	1	1	0	-1	-1
8/12 1500	1	1	0	-1	-2
8/12 1600	0	0	0	-2	-2
8/12 1700	1	1	0	-1	-1
8/12 1800	2	0	2	0	15
8/12 1900	2	2	2	1	15
8/12 2000	0	0	2	2	15
8/12 2100	2	0	4	3	15
8/12 2200	2	2	4	4	15
8/12 2300	2	0	6	5	15
8/13 0000	1	1	6	6	15
8/13 0100	0	0	6	7	15
8/13 0200	1	0	7	8	15
8/13 0300	0	0	7	9	15
8/13 0400	0	0	7	10	15
8/13 0500	0	0	7	11	15
8/13 0600	0	0	7	12	15
8/13 0700	0	2	5	13	15
8/13 0800	0	1	4	14	15
8/13 0900	0	2	2	15	15
8/13 1000	1	3	0	-1	-2
8/13 1100	0	0	0	-2	-2
8/13 1200	1	1	0	-1	-1

Time	Arv	Dpt	NetInv	Streak	Adjust
8/11 1800	0	0	0	-1	-1
8/11 1900	2	2	0	-1	-2
8/11 2000	0	0	0	-2	-2
8/11 2100	1	0	1	0	36
8/11 2200	2	0	3	1	36
8/11 2300	2	0	5	2	36
8/12 0000	1	0	6	3	36
8/12 0100	0	0	6	4	36
8/12 0200	1	0	7	5	36
8/12 0300	0	0	7	6	36
8/12 0400	0	0	7	7	36
8/12 0500	0	0	7	8	36
8/12 0600	0	0	7	9	36
8/12 0700	0	2	5	10	36
8/12 0800	0	1	4	11	36
8/12 0900	0	2	2	12	36
8/12 1000	0	1	1	13	36
8/12 1100	2	2	1	14	36
8/12 1200	1	1	1	15	36
8/12 1300	0	0	1	16	36
8/12 1400	1	1	1	17	36
8/12 1500	1	1	1	18	36
8/12 1600	0	0	1	19	36
8/12 1700	1	1	1	20	36
8/12 1800	2	2	1	21	36
8/12 1900	2	2	1	22	36
8/12 2000	0	0	1	23	36
8/12 2100	2	0	3	24	36
8/12 2200	2	2	3	25	36
8/12 2300	2	0	5	26	36
8/13 0000	1	1	5	27	36
8/13 0100	0	0	5	28	36
8/13 0200	1	0	6	29	36
8/13 0300	0	0	6	30	36
8/13 0400	0	0	6	31	36
8/13 0500	0	0	6	32	36
8/13 0600	0	0	6	33	36
8/13 0700	0	2	4	34	36
8/13 0800	0	1	3	35	36
8/13 0900	0	2	1	36	36
8/13 1000	1	3	-1	-1	-2
8/13 1100	0	0	-1	-2	-2
8/13 1200	1	1	-1	-1	-1

Figure B–6: Inventory Vectors for Third Shake Out showing the effect of delaying five flights, three for approximately one hour each and two for approximately two hours each

APPENDIX C

C. ESTIMATED AVERAGE (AND ROOT-MEAN-SQUARED) LOCATION ERROR PER TARGET FOR OPTIMAL TOURS

People can distinguish visually when a fleet of UAVs have reasonably partitioned a set of targets into compact, balanced subsets. However, it is more difficult to distinguish when a negotiation mechanism has converged to a good solution versus simply stalling out at a sub-optimal solution. For a set of targets scattered uniformly in space, we derive estimated average and root-mean-squared (RMS) location errors per target to be used as a baseline for the experimental analysis.

Given N points distributed uniformly in the unit square, Beardwood, *et al.* [BHH59] derived an asymptotic result. They showed that the expected ratio of the optimal TSP tour length through all N points to $\sqrt{N}$ approaches a limiting constant C as $N \to \infty$. Johnson, *et al.* estimate $C = 0.7124 \pm 0.0002$ in the limit [JMR96]. However, for $N < 1{,}000$, they show that 0.75 is a better estimate.

We use this approximation as the basis for estimating the average and root-mean-squared (RMS) location error per target associated with M UAVs (with speed u) servicing N targets (moving by random walk with step size v) over an area A. This derivation assumes that the UAVs have partitioned the targets into compact, balanced subsets and each UAV has constructed an optimal tour for its subset of targets.

Let the optimal tour length, L, associated with a single UAV servicing N targets over an area A be estimated by

$$L = 0.75 \cdot \sqrt{A} \cdot \sqrt{N}\ . \qquad \text{(C.1)}$$

Extending this to M UAVs that divide the surveillance area and target set, the estimated optimal tour length per UAV is

$$L = 0.75 \cdot \sqrt{\frac{A}{M}} \cdot \sqrt{\frac{N}{M}} = 0.75 \cdot \frac{\sqrt{AN}}{M}. \tag{C.2}$$

Note that the sum of the tour lengths across all UAVs in equation (C.2) is equal to the single UAV tour length in equation (C.1). Using equation (C.2) and the UAV speed *u*, the expected time between target visits, *T*, is equal to

$$T = \frac{L}{u} = 0.75 \cdot \frac{\sqrt{AN}}{Mu}. \tag{C.3}$$

Next, we compute the expected location error over the time interval [0, *T*]. The location error is equal to the distance between the current target location and the last known target location. For the Pearson random walk model with step size *v*, the Central Limit Theorem can be used to show that the expected distance from the initial position after *t* time periods is $v\sqrt{t}$ (see Hughes [Hug95] for one such derivation).

It follows that the expected distance from the initial target position averaged over the time interval [0, *T*] equals

$$d_{Avg} = \frac{1}{T}\int_0^T v\sqrt{t}\ dt = \frac{v}{T}\left[\frac{t^{3/2}}{3/2}\right]_0^T = \frac{2v}{3T}\left(T^{3/2}\right) = \frac{2}{3}v\sqrt{T}. \tag{C.4}$$

If we assume optimal tours and optimal assignments of targets to UAVs, then we can estimate the expected location error over time per target by combining equations (C.3) and (C.4),

$$d_{Avg} = \frac{2}{3}v\sqrt{0.75 \cdot \frac{\sqrt{AN}}{Mu}} \approx 0.58v \cdot \sqrt[4]{\frac{AN}{M^2u^2}}. \tag{C.5}$$

We can compute the RMS location error over this time interval similarly. For the Pearson random walk model with step size *v*, the expected squared distance from the initial position after *t* time periods is v^2t. The RMS distance from the initial target position over the time interval [0, *T*] equals

$$d_{RMS} = \sqrt{\frac{1}{T}\int_0^T v^2t\ dt} = \sqrt{\frac{v^2}{T}\left[\frac{t^2}{2}\right]_0^T} = v\sqrt{\frac{T^2-0}{2T}} = \frac{v}{\sqrt{2}}\sqrt{T}. \tag{C.6}$$

Combining equations (C.3) and (C.6), the estimated RMS location error per target over time, assuming optimal tours and target assignments, is

$$d_{RMS} = \frac{v}{\sqrt{2}} \sqrt{0.75 \cdot \frac{\sqrt{AN}}{Mu}} \approx 0.61 v \cdot \sqrt[4]{\frac{AN}{M^2 u^2}} \,. \tag{C.7}$$

The average and RMS error estimates differ by about six percent.

We can modify these derivations to handle a different initial target allocation for which the UAVs construct optimal tours based on a random assignment of targets and cannot perform target swapping. We call this variation the “No Swap” case. Equation (C.2) is similar, except the $\sqrt{A/M}$ term is replaced with $\sqrt{A}$ because each UAV has its tour across the entire area rather than a partitioned subset of space. The derivations continue as before, leading to the final results

$$d_{Avg}^{NoSwap} \approx 0.58 v \cdot \sqrt[4]{\frac{AN}{Mu^2}} \text{ and} \tag{C.8}$$

$$d_{RMS}^{NoSwap} \approx 0.61 v \cdot \sqrt[4]{\frac{AN}{Mu^2}} \,. \tag{C.9}$$

APPENDIX D

D. DERIVATION OF COOPERATIVE SCORING RULE

In this section, we derive a rule for determining whether a proposed swap between two UAVs is beneficial to the system with respect to minimizing the sum of the squared location error across all targets.

Consider the problem of $M = 2$ UAVs splitting a set of N targets. Let J_1 and J_2 be the set of targets owned by UAV 1 and UAV 2, respectively, such that $|J_1| + |J_2| = N$. The UAVs will exchange individual targets with each other, changing the composition of the sets J_1 and J_2 over time.

The system goal at a particular time is to minimize $\Sigma_{j=1}^{N} d_j^2$, where d_j^2 is the squared target location error of target j at that time. However, UAVs cannot measure this error directly because it requires knowing the actual target locations at that time. Instead, the UAVs will estimate this error based on the Pearson random walk model and the time since the target was last detected.

We start by estimating the sum of the squared errors for the targets owned by UAV 1. Let $t_{(1)} > t_{(2)} > t_{(3)} > \ldots > t_{(|J1|)}$ be the times since last detection for each of the targets owned by UAV 1. We assume that this indexing also represents the order in which the targets will be visited. Let l_1 be the current length of UAV 1's tour. Then we can estimate $t_{(j)}$ by

$$t_{(j)} = \frac{l_1}{u} \cdot \left(\frac{|J_1| + ½ - j}{|J_1|} \right) \text{ for } j = 1,\ 2,\ ...,\ |J_1| \tag{D.1}$$

Intuitively, the idea is that if a tour with $|J_1| = 10$ targets has a cycle time of $l_1/u = 100$, then the estimated time since last detection for each of the targets is 95, 85, 75, …, 5. We choose this midpoint between target visits because the errors just before a detection are artificially high and the errors just after a detection are artificially low.

For the Pearson random walk model with step size *v*, the expected squared distance from the initial target position after *t* time periods is v^2t [Hug95]. Thus, the expected sum of squared location errors from UAV 1's targets is

$$E\left[\sum_{j=1}^{|J_1|} d_{(j)}^2\right] = \sum_{j=1}^{|J_1|} E\left[d_{(j)}^2\right] = \sum_{j=1}^{|J_1|} v^2 t_{(j)} = v^2 \sum_{j=1}^{|J_1|} t_{(j)} \tag{D.2}$$

Combining equations (D.1) and (D.2) and then simplifying, we get

$$\begin{aligned}
E\left[\sum_{j=1}^{|J_1|} d_{(j)}^2\right] &= v^2 \sum_{j=1}^{|J_1|} \frac{l_1}{u} \cdot \left(\frac{|J_1| + ½ - j}{|J_1|}\right) \\
&= v^2 \cdot \frac{l_1}{u} \sum_{j=1}^{|J_1|} \cdot \left(1 + \frac{1}{2 \cdot |J_1|} - \frac{j}{|J_1|}\right) \\
&= v^2 \cdot \frac{l_1}{u} \cdot \left(|J_1| + \frac{|J_1|}{2 \cdot |J_1|} - \frac{1}{|J_1|} \sum_{j=1}^{|J_1|} j\right) \\
&= v^2 \cdot \frac{l_1}{u} \cdot \left(|J_1| + \frac{1}{2} - \frac{1}{|J_1|} \cdot \frac{\left(|J_1| + 1\right) \cdot |J_1|}{2}\right) \\
&= v^2 \cdot \frac{l_1}{u} \cdot \left(|J_1| + \frac{1}{2} - \frac{\left(|J_1| + 1\right)}{2}\right) \\
&= v^2 \cdot \frac{l_1}{u} \cdot \left(|J_1| + \frac{1}{2} - \frac{|J_1|}{2} - \frac{1}{2}\right) = \frac{v^2}{2u} \cdot l_1 \cdot |J_1|
\end{aligned}$$

The estimated sum of squared errors can then be written as

$$\sum_{j=1}^{N} d_j^2 \approx \frac{v^2}{2u} \cdot \left(l_1 \cdot |J_1| + l_2 \cdot |J_2|\right). \tag{D.3}$$

Since *u* and *v* are constant throughout the simulation and do not depend on the target assignments, we drop the multiplier in the decision rule. Consider a set of target assignments J_1 and J_2 and tour lengths of l_1 and l_2 for two UAVs. The cooperative decision rule for evaluating swap proposals is that a swap proposal that leads to assignments J_1' and J_2' and tour lengths l_1' and l_2' will be accepted only if

$$l_1' \cdot |J_1'| + l_2' \cdot |J_2'| < l_1 \cdot |J_1| + l_2 \cdot |J_2|. \tag{D.4}$$

APPENDIX E

E. PROPERTIES OF RANDOM WALK MOTION MODEL

In this appendix, we derive several properties of the Pearson random walk model for target motion, including derivations of the spatial distribution as a function of time and the probability transition process associated with diffusion on the search area hexagonal grid.

E.1 SPATIAL DISTRIBUTION FOR PEARSON RANDOM WALK MODEL

For a Pearson random walk model, at each time step, the target takes a fixed step length *v* in a uniformly random direction *θ*. In this section, we derive the statistical properties of this random walk process as a function of time, and then show how to discretize the 2-D spatial distribution associated with this process into the hexagonal grid used for the search area.

E.1.1 Statistical properties of Pearson random walk process

Mean. Consider a single random step of length *v* (see Figure E–1). Let *Θ* be a random variable drawn from *U* [0, 2*π*] and *X* = *v* cos(*Θ*) be a random variable representing the horizontal component of the step. The expected value of *X* is

$$\begin{aligned} E[X] &= \int_0^{2\pi} x \cdot \left(\frac{1}{2\pi} d\theta \right) = \int_0^{2\pi} \left(v\cos\theta \right) \cdot \left(\frac{1}{2\pi} d\theta \right) = \frac{v}{2\pi} \int_0^{2\pi} \cos\theta \, d\theta \\ &= \frac{2v}{2\pi} \int_0^{\pi} \cos\theta \, d\theta = \frac{v}{\pi} \left[\sin\theta \right] \Big|_0^{\pi} = \frac{v}{\pi} \left[\sin\pi - \sin 0 \right] = 0. \end{aligned} \tag{E.1}$$

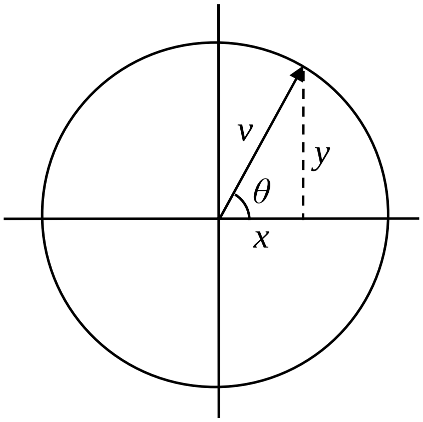


Figure E–1: Depiction of a single step of the Pearson random walk process

Variance. To compute the variance, we need to derive an identity. We start with

$$2\pi \;=\; \int_0^{2\pi} 1\cdot d\theta \;=\; \int_0^{2\pi}\left(\sin^2\theta+\cos^2\theta\right)d\theta \;=\; \int_0^{2\pi}\left(\sin^2\theta\right)d\theta+\int_0^{2\pi}\left(\cos^2\theta\right)d\theta\,.$$

Since both of these terms are equal, then we have the following identity

$$\pi \;=\; \int_0^{2\pi}\left(\cos^2\theta\right)d\theta\,. \tag{E.2}$$

The variance of *X* is

$$\begin{aligned} Var[X] \;&=\; E[X^2]-E[X]^2 \;=\; E[X^2]-0 \\ &=\; \frac{1}{2\pi}\int_0^{2\pi}\left(v\cdot\cos\theta\right)^2\,d\theta \;=\; \frac{v^2}{2\pi}\int_0^{2\pi}\cos^2\theta\,d\theta \;=\; \frac{v^2\pi}{2\pi} \;=\; \tfrac{1}{2}v^2. \end{aligned} \tag{E.3}$$

To generalize these results to multiple steps, let $\hat{X}_t = \sum_{i=1}^{t} X_i$ where X_i are independent, identically distributed random variables of the form described above such that $E[X_i] = 0$ and $Var[X_i] = \tfrac{1}{2}v^2$. Then $E[\hat{X}_t] = 0$ and $Var[\hat{X}_t] = \tfrac{1}{2}v^2 t$. By the Central Limit Theorem, as *t*→∞, the distribution of the random variable $\hat{X}_t \big/ \sqrt{t}$ converges to the Normal Distribution with mean 0 and variance $\tfrac{1}{2}v^2$.

Probability Density Function. By symmetry, one can make similar arguments for the random variable $Y = v\sin(\Theta)$ to determine that $E[\hat{Y}_t] = 0$, $Var[\hat{Y}_t] = \tfrac{1}{2}v^2 t$, and as *t*→∞, the distribution of the random variable $\hat{Y}_t \big/ \sqrt{t}$ converges to the Normal Distribution with mean 0 and variance $\tfrac{1}{2}v^2$.

Although X_1, X_2, …, X_t are conditionally independent random variables and Y_1, Y_2, …, Y_t are conditionally independent, X_i and Y_i are not conditionally independent because they share a dependence on Θ_i. However, using an offset argument, X_i and Y_{i+1} can be shown to be conditionally independent for i = 1, 2, …, t–1.

In the limit as $t \to \infty$, $\hat{X}_t$ and $\hat{Y}_t$ become statistically uncorrelated and the distribution of the position $\left(\hat{x}_t, \hat{y}_t\right)$ converges to the Bivariate Normal Distribution, which has the following probability density function

$$f(\hat{x}_t, \hat{y}_t) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\left(\frac{\hat{x}_t - \mu_x}{\sigma_x}\right)^2 + \left(\frac{\hat{y}_t - \mu_y}{\sigma_y}\right)^2\right)}. \tag{E.4}$$

In the case of the Pearson random walk model, $\mu_x = \mu_y = 0$ and $\sigma_x = \sigma_y = v\sqrt{\tfrac{1}{2}t}$, which we will relabel for now as σ_t. Substituting these values, equation (E.4) can be rewritten as

$$f(\hat{x}_t, \hat{y}_t) = \frac{1}{2\pi\sigma_t^2} e^{-\frac{\hat{x}_t^2 + \hat{y}_t^2}{2\sigma_t^2}}.$$

Finally, we transform the distribution to a polar coordination representation by defining the radial component to be $\hat{r}_t^2 \equiv \hat{x}_t^2 + \hat{y}_t^2$. The probability density function becomes

$$f(\hat{r}_t, \theta) = \frac{1}{2\pi\sigma_t^2} e^{-\frac{1}{2}\left(\frac{\hat{r}_t}{\sigma_t}\right)^2}. \tag{E.5}$$

Cumulative Distribution Function. Next, we derive the cumulative distribution function for the random walk model. In particular, we want to know the probability that a target moving with step size v is within distance d_t of its initial position after t steps. Let $\hat{R}_t^2 \equiv \hat{X}_t^2 + \hat{Y}_t^2$ be the random variable describing the radius distance after t steps. Then

$$\begin{aligned} F\left(d_t\right) = P\left[\hat{R}_t \le d_t\right] &= \int_0^{d_t}\int_0^{2\pi} \frac{1}{2\pi\sigma_t^2} e^{-\frac{1}{2}\left(\frac{r}{\sigma_t}\right)^2} r\,dr\,d\theta \\ &= \frac{1}{2\pi}\int_0^{d_t} e^{-\frac{1}{2}\left(\frac{r}{\sigma_t}\right)^2}\left(\tfrac{r}{\sigma_t}\right)\left(\tfrac{dr}{\sigma_t}\right)\int_0^{2\pi} d\theta = \int_0^{d_t} e^{-\frac{1}{2}\left(\frac{r}{\sigma_t}\right)^2}\left(\tfrac{r}{\sigma_t}\right)\left(\tfrac{dr}{\sigma_t}\right) \end{aligned}$$

Applying the substitution $u = r/\sigma_t$ and rescaling the bounds of integration by defining $k = d_t/\sigma_t$, we have the following cumulative distribution function (CDF),

$$F\left(d_t\right) = P\left[\hat{R}_t \le k\sigma_t\right] = \int_0^k e^{-\frac{1}{2}u^2} u\, du = -e^{-\frac{1}{2}u^2}\Big|_0^k = 1 - e^{-\frac{1}{2}k^2} = 1 - e^{-\frac{1}{2}\left(\frac{d_t}{\sigma_t}\right)^2}. \qquad \text{(E.6)}$$

E.1.2 Converting continuous spatial distribution to discretized hexagon cells

In this section, we discretize the continuous probability density function of a target performing a Pearson random walk for *t* steps onto a tiling of regular hexagons. Each cell would then contain the probability that the target is contained within that cell.

First, we approximate the regular hexagon with edge length *s* by a circle with radius r_E that has the same area (see Figure E–2). A regular hexagon is a tiling of six equilateral triangles. The area of an equilateral triangle with edge length *s* is

$$Area(triangle) = \frac{1}{2}(base)(height) = \frac{1}{2}s\left(\frac{\sqrt{3}}{2}s\right) = \frac{\sqrt{3}}{4}s^2.$$

The area of a regular hexagon with edge-length *s* is therefore

$$Area(hexagon) = 6 \times Area(triangle) = \frac{6\sqrt{3}}{4}s^2 = \frac{3\sqrt{3}}{2}s^2.$$

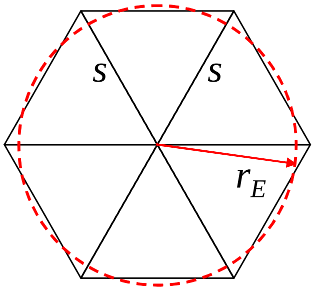


Figure E–2: Relationship between regular hexagon and circle with equivalent area

To compute r_E, we set the area of the circle equal to the area of the regular hexagon with side length *s*. Solving for r_E, we get

$$\begin{aligned} \pi r_E^2 &= \frac{3\sqrt{3}}{2}s^2 \\ r_E^2 &= \frac{3\sqrt{3}}{2\pi}s^2 \qquad \text{(E.7)} \\ r_E &= s\sqrt{\frac{3\sqrt{3}}{2\pi}} \approx 0.91\, s \end{aligned}$$

Consider a hexagonal cell whose center is distance *d* from the center of the distribution (see Figure E–3). In order to compute the probability that the target is in that cell, we start by computing the probability that the target is within the proper distance. That is, we calculate the probability that the target is contained within a circle of radius ($d–r_E$) and subtract that from the probability that the target is contained within a circle of radius ($d+r_E$). This gives us the probability that the target is contained within a shell of radial width $2r_E$.

$$F\left(d+r_E\right)-F\left(d-r_E\right) = 1-e^{-\frac{1}{2}\left(\frac{d+r_E}{\sigma_t}\right)^2} -1+e^{-\frac{1}{2}\left(\frac{d-r_E}{\sigma_t}\right)^2} = e^{-\frac{1}{2}\left(\frac{d-r_E}{\sigma_t}\right)^2} -e^{-\frac{1}{2}\left(\frac{d+r_E}{\sigma_t}\right)^2} \qquad \text{(E.8)}$$

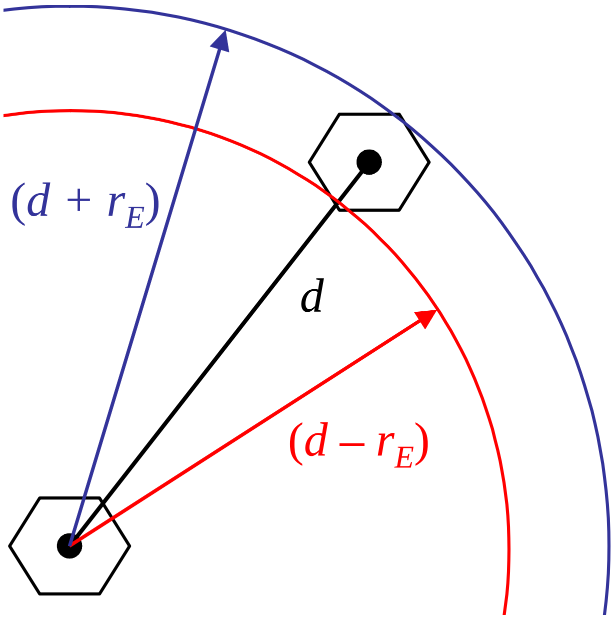


Figure E–3: Illustration for computing the probability of a target being in a cell

We will assume that this probability is spread uniformly throughout the shell, which is not true but serves as a reasonable approximation. We assign to the cell a fraction of that shell probability, proportional to the ratio of the hexagonal area to the area of the shell. That is, we can compute the probability, *p*, of the target being in the cell to be

$$p = \left(shell\ probability\right)\cdot\left(\frac{area\ of\ hexagon}{area\ of\ shell}\right)$$

$$p = \left[F\left(d+r_E\right)-F\left(d-r_E\right)\right]\cdot\left(\frac{\frac{3\sqrt{3}}{2}s^2}{\pi\left(d+r_E\right)^2-\pi\left(d-r_E\right)^2}\right)$$

$$p \; = \left[F\left(d + r_E\right) - F\left(d - r_E\right) \right] \cdot \left(\frac{\frac{3\sqrt{3}}{2} s^2}{\pi d^2 + 2\pi d r_E + \pi r_E^2 - \pi d^2 + 2\pi d r_E - \pi r_E^2} \right)$$

$$p \; = \left[F\left(d + r_E\right) - F\left(d - r_E\right) \right] \cdot \left(\frac{\frac{3\sqrt{3}}{2} s^2}{4\pi d r_E} \right)$$

$$p \; = \left[e^{-\frac{1}{2}\left(\frac{d - r_E}{\sigma_t}\right)^2} - e^{-\frac{1}{2}\left(\frac{d + r_E}{\sigma_t}\right)^2} \right] \cdot \frac{3\sqrt{3}}{8\pi d r_E} s^2 \tag{E.9}$$

Note that for the special case of the hexagonal cell in the center of the distribution, the cell probability is equal to $F(r_E)$. Due to the assumption of uniform weight throughout the shell, we need to renormalize the cell probabilities to ensure that the total probability across all cells equals one.

E.2 Derivation of Probability Transition Motion Model

Given that a target is in a particular hexagonal cell, we need to compute the probability of the target leaving the cell in the next step of fixed length v. In this section, we derive an approximation for this transition probability and use it as the basis for the motion model used to update the target prior distribution on location.

E.2.1 Computing the transition probability for fixed step size *v*

As before, we will approximate the hexagon cell as a circle with equivalent radius. To simplify notation, let r be the radius of this equivalent circle. Assume a target is placed within this circle according to a uniform distribution. Instead of treating the target as a discrete particle, we will consider the transition of the entire distribution.

Suppose the entire distribution takes a fixed step v in a random direction, such as shown in Figure E–4. What proportion of the distribution falls outside the original circle (the shaded area Q)? This is the same as the probability of a target transitioning out of the circle in one step, which we will call q.

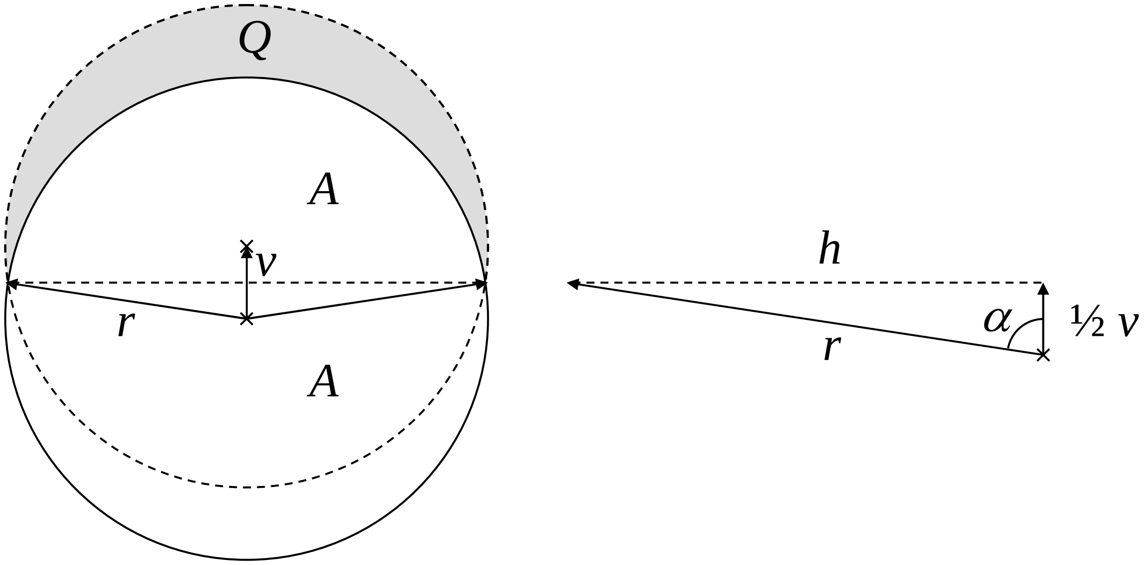


Figure E–4: Geometry associated with a fixed step size, *v*

The overlap between the two circles (before and after the fixed step) has symmetric top and bottom halves, each of which has area *A*. We can decompose the total area of the circle into this overlap and the area that falls outside the original circle by

$$\pi r^2 \ = \ Q + A + A \, .$$

Dividing through by πr^2, we can transform these areas into probabilities,

$$1 \ = \ \frac{Q}{\pi r^2} + \frac{2A}{\pi r^2} \ = \ q + \frac{2A}{\pi r^2}$$

$$q \ = \ 1 - \frac{2A}{\pi r^2} \, . \tag{E.10}$$

Returning to Figure E–4, we can relate the area of the circular wedge with sweep angle 2α to the area *A* and the two triangles with base *h* and height *v*/2.

$$\left(\frac{2\alpha}{2\pi}\right)\pi r^2 \ = \ A + 2 \cdot \left(\tfrac{1}{2} h \left(\tfrac{1}{2} v\right)\right)$$

$$\alpha r^2 \ = \ A + \tfrac{1}{2} h v$$

$$\alpha r^2 - \tfrac{1}{2} h v \ = \ A \tag{E.11}$$

Substituting the value of *A* in equation (E.11) into equation (E.10), we get

$$q \;=\; 1-\frac{2\left(\alpha r^2-\tfrac{1}{2}hv\right)}{\pi r^2} \;=\; 1-\frac{2\alpha r^2-hv}{\pi r^2} \;=\; 1-\frac{2\alpha}{\pi}+\frac{hv}{\pi r^2}\,.$$

Using trigonometric substitutions, we can express q in terms of α alone.

$$q \;=\; 1-\frac{2\alpha}{\pi}+\left(\frac{h}{r}\right)\left(\frac{v}{\pi r}\right) \;=\; 1-\frac{2\alpha}{\pi}+\frac{2}{\pi}\left(\frac{h}{r}\right)\left(\frac{v}{2r}\right) \;=\; 1-\frac{2\alpha}{\pi}+\frac{2}{\pi}\cos\alpha\sin\alpha \qquad \text{(E.12)}$$

$$q \;=\; 1-\frac{1}{\pi}\left(2\alpha-\sin 2\alpha\right)\ \text{ where } \alpha=\cos^{-1}\left(\frac{v}{2r}\right) \qquad \text{(E.13)}$$

The limiting behavior for the two extreme step size cases (one as v approaches zero, and the other as v approaches 2r) is as expected:

$$\lim_{v\to 0} q \;=\; \lim_{\alpha\to\pi/2} q \;=\; 1-\frac{1}{\pi}\left(\pi-\sin\pi\right) \;=\; 1-\left(1-0\right) \;=\; 0\text{, and}$$

$$\lim_{v\to 2r} q \;=\; \lim_{\alpha\to 0} q \;=\; 1-\frac{1}{\pi}\left(0-\sin 0\right) \;=\; 1-\left(0-0\right) \;=\; 1\,.$$

In Figure E–5, we show the functional form of the transition probability as a function of the step size v divided by the equivalent cell radius r. The function is roughly linear for $v < r$. In the next section, we perform a Taylor expansion that shows that the linear term dominates the approximation.

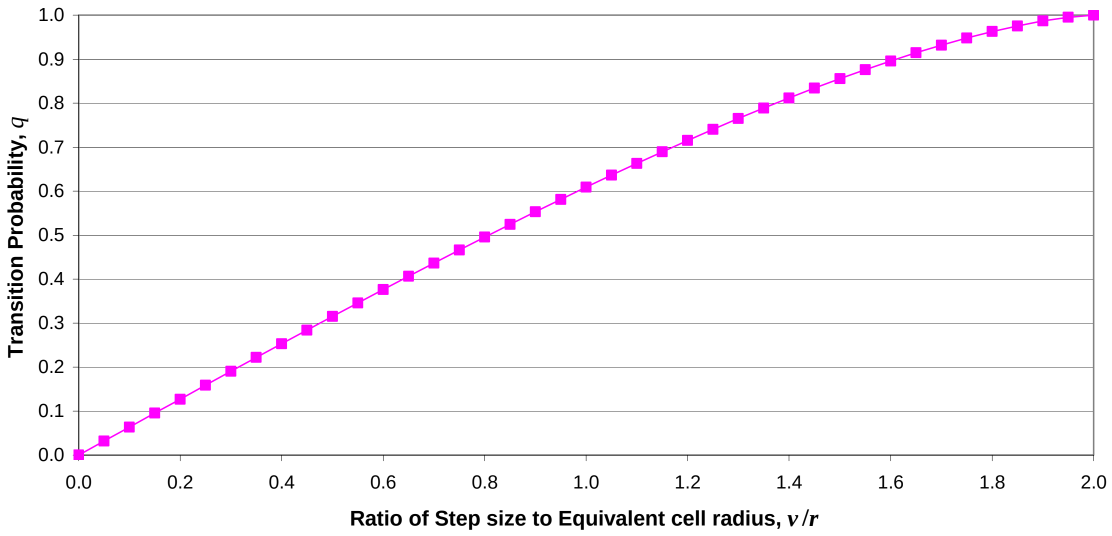


Figure E–5: Plot of transition probability *q* as a function of the scaled step size

We can now define the motion update to transform the prior distribution based on the dynamic motion of each target following a Pearson random walk process. Define $p_i\,(t–1)$ to be the probability that a target is in cell i at time t–1 based on equation (E.9). The goal is to transform that prior into the time t distribution $p_i^-(t)$ by applying the motion update.

Let us assume as before that a target is positioned at random uniformly within a particular cell. If the target leaves that cell in the next step, then it is equally likely to step into any of the six immediate neighboring cells. In addition, we have shown that the target will remain in the original cell with probability 1–q. Thus, the motion update equation can be expressed as

$$p_i^-(t) = \sum_j q\left(i \mid j\right) \cdot p_j\left(t-1\right), \text{ where} \tag{E.14}$$

$$q(i \mid j) = \begin{cases} 1-q & \text{if } \; j = i \\ q/6 & \text{if } \; j \text{ is an immediate neighbor of } i \; . \\ 0 & \text{otherwise} \end{cases}$$

The primary drawback of this transition model is the assumption that the target location follows a uniform distribution within the cell, and thus, that the target is equally likely to jump to any of the six neighbors when it leaves its cell. Consider the case in which $v << r_E$ and a target takes a step from cell A to cell B. Due to the small step size, the target's position within cell B is likely to be closer to cell A than to the other neighbors of cell B. Consequently, if the target takes a step in the next time increment that causes it to leave cell B, then it is more likely to return to cell A than to move to any other neighbor of cell B. This violates the uniformity assumption implicit in equation (E.14).

Another consequence of assuming that the transition model distributes the cell weights evenly to the neighbors is that the spatial variance using this transition model increases slightly faster than the 2-D Gaussian described in section E.1. Despite this drawback, we choose to keep the uniformity assumption. Doing so leads to a closed-form expression for the transition probability and retains the Markov (memory-less) assumption regarding the target motion. Relaxing this assumption, on the other hand, makes deriving a reasonable transition model much more difficult without necessarily leading to a significant improvement in the operational search effectiveness.

In the next section, we derive a Taylor expansion of the transition probability function in equation (E.12) in order to understand better the functional form of this expression.

E.2.2 Taylor expansion of the transition probability function

The transition probability function approximation using the Taylor expansion starts with equation (E.12) and uses the identity cos^{-1}(x) = π/2 – sin^{-1}(x). To simplify the expressions, define the variable $\overline{v} = v/2r$ such that $\alpha = \cos^{-1}\overline{v}$. Then we can write

$$\begin{aligned}
q &= 1 - \frac{2\alpha}{\pi} + \frac{2}{\pi}\cos\alpha\sin\alpha \\
&= 1 - \frac{2}{\pi}\left(\frac{\pi}{2} - \sin^{-1}\overline{v}\right) + \frac{2}{\pi}\cos\left(\cos^{-1}\overline{v}\right)\sin\left(\cos^{-1}\overline{v}\right) \\
&= 1 - \left(1 - \frac{2}{\pi}\sin^{-1}\overline{v}\right) + \frac{2}{\pi}\left(\overline{v}\right)\sin\left(\cos^{-1}\overline{v}\right) \\
&= \frac{2}{\pi}\left[\sin^{-1}\overline{v} + \overline{v}\sin\left(\cos^{-1}\overline{v}\right)\right].
\end{aligned}$$

Next, we substitute the identity $\sin\left(\cos^{-1}x\right) = \sqrt{1-x^2}$ to yield

$$q = \frac{2}{\pi}\left[\sin^{-1}\overline{v} + \overline{v}\sqrt{1-\overline{v}^2}\right]. \qquad \text{(E.15)}$$

There are two Taylor expansions that we will use:

$$\sin^{-1}x = x + \frac{1}{3!}x^3 + \frac{3^2}{5!}x^5 + \ldots = x + \frac{1}{6}x^3 + \frac{3}{40}x^5 + \ldots \text{ and}$$

$$\sqrt{1-x^2} = 1 - \frac{1}{2}x - \frac{1}{6}x^2 + \ldots.$$

Substituting these expansions into equation (E.15) yields

$$\begin{aligned}
q &\approx \frac{2}{\pi}\left[\overline{v} + \frac{1}{6}\overline{v}^3 + \frac{3}{40}\overline{v}^5 + \overline{v}\left(1 - \frac{1}{2}\overline{v}^2 - \frac{1}{6}\overline{v}^4\right)\right] \\
&= \frac{2}{\pi}\left[\overline{v} + \frac{1}{6}\overline{v}^3 + \frac{3}{40}\overline{v}^5 + \overline{v} - \frac{1}{2}\overline{v}^3 - \frac{1}{6}\overline{v}^5\right] \\
&= \frac{2}{\pi}\left[2\overline{v} + \left(\frac{1}{6} - \frac{1}{2}\right)\overline{v}^3 + \left(\frac{3}{40} - \frac{1}{6}\right)\overline{v}^5\right] \\
&= \frac{2}{\pi}\left[2\overline{v} - \frac{1}{3}\overline{v}^3 - \frac{11}{120}\overline{v}^5\right]
\end{aligned}$$

Replacing $\overline{v}$ with $v/2r$ and simplifying, we get our final result,

$$q \;\approx\; \frac{2}{\pi}\left[2\left(\frac{v}{2r}\right)-\frac{1}{3}\left(\frac{v}{2r}\right)^3-\frac{11}{120}\left(\frac{v}{2r}\right)^5\right]$$

$$= \frac{4}{2\pi}\left(\frac{v}{r}\right)-\frac{2}{3\cdot 8\pi}\left(\frac{v}{r}\right)^3-\frac{22}{120\cdot 32\pi}\left(\frac{v}{r}\right)^5$$

$$= \frac{2}{\pi}\left(\frac{v}{r}\right)-\frac{1}{12\pi}\left(\frac{v}{r}\right)^3-\frac{11}{1920\pi}\left(\frac{v}{r}\right)^5 .$$

Note that we have expressed this equation with respect to the rescaled step size v/r. Numerically, this simplifies as

$$q \;\approx\; 0.637\left(\frac{v}{r}\right)-0.027\left(\frac{v}{r}\right)^3-0.002\left(\frac{v}{r}\right)^5 . \qquad \text{(E.16)}$$

Clearly, the linear term dominates the transition probability, especially for $v < r$. Figure E–6 shows the close fit between the analytical form of the transition probability in equation (E.13) and the fifth-degree Taylor expansion in equation (E.16).

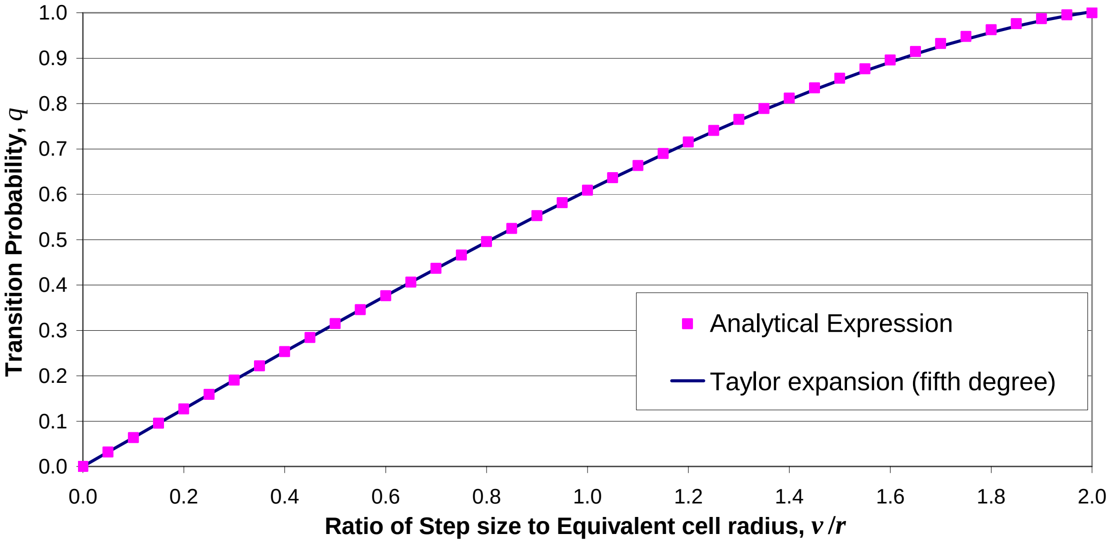


Figure E–6: Fifth-degree Taylor expansion fits analytical transition probability